VIZUALNO UČENJE PROSTORSKO-ČASOVNEGA MODELA CIKLIČNEGA ČLOVEŠKEGA GIBANJA ZA RAZPOZNAVANJE IN SLEDENJE

Miha Peternel

MAGISTRSKA NALOGA
predložena
Fakulteti za računalništvo in informatiko
Univerze v Ljubljani
kot delna izpolnitev pogoja za pridobitev naslova
magister računalništva in informatike

Ljubljana, 2004

Mentor:
prof. dr. Aleš Leonardis

VISUAL LEARNING OF A SPATIO-TEMPORAL MODEL OF CYCLIC HUMAN LOCOMOTION FOR RECOGNITION AND TRACKING

Miha Peternel

MASTER'S THESIS
submitted to the
Faculty of Computer and Information Science of
University of Ljubljana
in partial fulfillment of the requirements for the
Master of Science Degree in Computer and Information Science

Ljubljana, 2004

Supervisor:
prof. dr. Aleš Leonardis

Magistrska naloga je bila izdelana pod mentorstvom prof. dr. Aleša Leonardisa in je last Fakultete za računalništvo in informatiko v Ljubljani. Za objavljanje in uporabo rezultatov magistrskega dela je potrebno soglasje zgoraj omenjene ustanove.

Povzetek

Vizualno sledenje je področje računalniškega vida, ki se ukvarja s tehnikami sledenja objektov v časovnem zaporedju slik. Te tehnike je možno uporabiti za sledenje ljudi, razpoznavanje značilnosti gibanja in razpoznavanje samih gibov.

Želeli bi razviti avtonomni sistem, ki bi se naučil lastnosti gibanja posameznikov iz množice učnih video posnetkov in nato uporabil naučeno znanje na novih posnetkih za sledenje in razpoznavanje.

Človeško telo lahko zavzame veliko število poz in gibov, kar tvori obsežno bazo učnih slik, zato potrebujemo model, ki kompaktno povzame gibanje iz zaporedja slik, a ne zanemari individualnih podrobnosti. Motiv je razvoj algoritma, ki določi parametre modela na podlagi posnetkov človeškega gibanja v delno nadzorovanem okolju in naučen model uporabi na novih posnetkih v poljubnem okolju za razpoznavanje ali sledenje.

V magistrski nalogi predstavljamo novo predstavitev cikličnega človeškega gibanja zasnovano na množici prostorsko-časovnih krivulj naključnih zasledovanih točk na površini človeka.

Pričnemo s postopkom za sledenje večih naključnih točk na človeku v video posnetku iz ene kamere, sledi metoda za določanje intervala ponavljanja, poravnavanja ponovitev in izločitev množice zveznih, fazno poravnanih prostorsko-časovnih krivulj. Analiziramo predstavitev cikličnih krivulj v prostoru glavnih komponent in izpostavimo lastnosti, ki so uporabne za prostorsko-časovno poravnavanje v sistemih za sledenje in razpoznavanje. Gostoto porazdelitve krivulj modeliramo z mešanico Gaussov s pomočjo postopka Expectation-Maximization. Za razpoznavanje uporabljamo oceno maksimalne posteriorne verjetnosti v kombinaciji z linearnim prilagajanjem podatkov.

Postopke smo testirali na bazi CMU MoBo z ugodnimi rezultati za razpoznavanje človeške identitete in tipa gibanja iz stranskih video posnetkov ene kamere. Zaključujemo z razpravo o rezultatih laboratorijskega in izvenlaboratorijskega testiranja in predlagamo razširitve metode.

II

Abstract

Visual tracking is a field of computer vision research that investigates techniques of object tracking in time sequences of images. These techniques can be applied to people tracking, gait recognition and activity recognition.

We envision an autonomous system which is able to learn appearance dynamics of humans from past video recordings and apply the information learned to assist recognition and tracking in new recordings.

Human body is a relatively complex articulate object with a multitude of possible poses and motions, making for a large database of learning images, therefore we need a model that compactly abstracts the motion in a video sequence, but doesn't discard individual characteristics. The main motive is the development of an algorithm which estimates the parameters of a model based on the video recordings of human locomotion in partially controlled environment and uses the learned model on new video recordings in arbitrary environment for recognition or tracking.

This thesis presents a novel representation of cyclic human locomotion based on a set of spatio-temporal curves of random tracked points on the surface of a person.

We start with an algorithm for tracking of multiple random points on a person in a monocular video sequence, followed by a method to determine the cycle interval, align repetitions and extract a set of continuous, phase aligned spatio-temporal curves. We analyze a PCA representation of the cyclic curves, pointing out properties of the representation which can be used for spatio-temporal alignment in tracking and recognition tasks. We model the curve distribution density by a mixture of Gaussians using Expectation-Maximization algorithm. For recognition, we use maximum *a posteriori* likelihood estimate combined with linear data adaptation.

We tested the algorithms on CMU MoBo database with favorable results for the recognition of people identity and locomotion mode from monocular video sequences captured from the side view. We conclude with discussion of results based on laboratory and outdoor testing and propose extensions of the method.

Acknowledgements

First I would like to thank Prof. Aleš Leonardis to provide mentorship, support, and motivation throughout the thesis development, as well as demanding and living up to high scientific standards.

I am grateful to Prof. Franc Solina for inviting me to work in the faculty setting. I would also like to thank him as well as Prof. Horst Bischof and Prof. Neža Mramor-Kosta to serve as members of the thesis committee.

I wish to thank all the members of Computer Vision Laboratory for providing a pleasant atmosphere and a constructive environment.

I would like to thank Ralph Gross for providing access to the CMU MoBo database.

I want to thank Sabina Murnik for assistance at video recording sessions.

Many thanks to Aleks Jakulin for countless intellectually stimulating debates and AI discussions that helped direct my search to some of the solutions in this thesis.

I want to express my gratitude to my family to support me during studies.

Finally I wish to thank the government of Slovenia to support my post-graduate studies through a Zois scholarship for gifted students.

Contents

Povzetek I
Abstract III
Acknowledgements V

List of Figures **XI**

List of Tables **XIII**

1 Introduction **1**
1.1 Motivation 1
1.2 General background 2
1.3 Problem statement 3
1.4 Thesis contributions 3
1.5 Thesis structure 3

2 Background and Related work **5**
2.1 Review of literature 5
2.2 Related work 6
2.3 Modeling human locomotion 9
2.4 Our approach 10

3 Probabilistic spatio-temporal model **13**
3.1 Overview 13
3.2 Spatio-temporal curve extraction 13
3.2.1 Moving object extraction 15
3.2.2 Point-tracking in a monocular sequence 15
3.2.3 Cycle detection and curve extraction 17
3.3 Learning of a probabilistic spatio-temporal model 20
3.4 Curve set in a PCA subspace 21

3.5 Subspace modeling of a spatio-temporal curve set 25
3.5.1 Properties of a spatio-temporal curve set of locomotion 25
3.5.2 Alternative spatio-temporal curve set representations 26
3.5.3 Gaussian mixture model of curve distribution 28
3.5.4 A probabilistic spatio-temporal model of locomotion 30
3.6 Recognition 30
3.6.1 Spatio-temporal alignment 31
3.6.2 Linear data adaptation 32
3.6.3 Classification 33
3.6.4 Application to recognition and tracking 34

4 Implementation **35**
4.1 Video input preprocessing 36
4.2 Moving object extraction 36
4.2.1 Global luminance normalization 36
4.2.2 Background subtraction 37
4.2.3 Shadow suppression 37
4.2.4 Morphological filtering 38
4.3 Point-tracking in a monocular sequence 38
4.3.1 A point-model for point tracking 38
4.3.2 Initialization 38
4.3.3 Tracking 39
4.3.4 Adding new random points 40
4.3.5 Point tracker output 41

5 Experimental Results **43**
5.1 CMU MoBo database 43
5.2 Learning 45
5.2.1 PCA decomposition and reconstruction 45
5.2.2 Analysis of parametric similarity 46
5.2.3 Analysis of subspace distribution similarity 54
5.2.4 Choosing the number of Gaussians 55
5.2.5 Expectation-Maximization convergence 57
5.3 Recognition 58
5.3.1 Principal component based phase alignment 58
5.3.2 Scale invariance 61

5.3.3 Approximate spatial alignment 62
5.3.4 Identification 63
5.3.5 Identity and Activity recognition 70
5.4 Testing in uncontrolled environment 72

6 Discussion **75**
6.1 Further possible extensions 75
6.2 Knowledge transfer 76
6.3 On extensions and alternative implementations 77
6.3.1 Distribution matching 77
6.3.2 Overcoming continuity requirement 78
6.4 Future work 79
6.5 Conclusion 80

A Principal component analysis **81**
A.1 Basics of PCA 81
A.2 Derivation of PCA 82
A.3 Properties of PCA 83

B Expectation-Maximization for Gaussian mixture models **87**
B.1 Basics of Expectation-Maximization 87
B.2 Derivation of EM for diagonal Gaussian mixture models 88
B.3 Initialization 91

C Razširjen povzetek v slovenščini **93**
C.1 Uvod 93
C.2 Naš pristop 94
C.3 Sledenje točk na gibajočem objektu v posnetku 95
C.4 Detekcija ciklov in izsejanje krivulj 96
C.5 Probabilistični prostorsko-časovni model 97
C.5.1 Množica krivulj v podprostoru glavnih komponent 97
C.5.2 Modeliranje porazdelitve krivulj z mešanico Gaussov 99
C.6 Razpoznavanje 100
C.6.1 Prostorsko-časovno poravnavanje 100
C.6.2 Klasifikacija 101
C.7 Eksperimenti 101
C.7.1 Učenje 102

C.7.2 Razpoznavanje 102
C.7.3 Testi v nenadzorovanem okolju 103
C.8 Razprava 104
C.9 Zaključek 104

Bibliography **107**

List of Figures

3.1 Overview of ST-curve extraction from a video sequence of locomotion . . 14
3.2 Original image, moving object extraction and point-tracking 16
3.3 Sample trajectories obtained by point-tracking of 10 random points 17
3.4 Voting accumulation vector for cycle start candidates 18
3.5 A set of phase aligned spatio-temporal curves 19
3.6 3D and 2D graphs of a spatio-temporal curve 20
3.7 Overview of learning of a spatio-temporal model of locomotion 21
3.8 Variance contained in the first D eigenvectors 22
3.9 The first 4 principal vectors 23
3.10 2D and 3D subspace mappings of ST-curve set 24
3.11 Overview of recognition of a locomotion pattern given a prior spatio-temporal model 31

4.1 Screenshot of the application during point tracking 35
4.2 Number of successfully tracked points throughout 25 fast walk sequences 40
4.3 Sample trajectories obtained by point-tracking of 10 random points 41

5.1 Modes and angles of experimental MoBo locomotion sequences 43
5.2 Cumulative share of variance contained in the principal eigenvalues . . . 46
5.3 The first 16 principal vectors of a fast walk learning sequence 47
5.4 Comparison of the principal vectors for different locomotion modes of a single person 48
5.5 The mean vectors of fast walk learning sequences 49
5.6 The 1st principal vectors of fast walk learning sequences 50
5.7 The 2nd principal vectors of fast walk learning sequences 51
5.8 The 3rd principal vectors of fast walk learning sequences 52
5.9 The 4th principal vectors of fast walk learning sequences 53
5.10 Scatter plots of prior and observed distributions of fast walk sequences . . 54

5.11 Illustration of Gaussian mixture models with varying number of Gaussians 55
5.12 Spatio-temporal curves of 15 Gaussian means 56
5.13 Convergence of expectation-maximization after 15 random initializations 57
5.14 Pairs of learned and observed 1st principal vectors 60
5.15 Effect of phase alignment on maximum *a posteriori* likelihood 60
5.16 Approximate remapping of ST-curve distributions in 2-dimensional subspace 63
5.17 Probable causes of recognition errors 64
5.18 Recognition results: fast walk 65
5.19 Recognition results: slow walk 66
5.20 Recognition results: incline walk 67
5.21 Recognition results: ball walk 68
5.22 Recognition results: slow walk - 45° back-view 69
5.23 Recognition results: Identity and Activity 71
5.24 Testing in uncontrolled environment 73

A.1 Typical eigenspectrum and energy of a set of spatio-temporal curves . . . 85

C.1 Originalna slika, izločanje gibajočega objekta in sledenje točk 95
C.2 Primeri trajektorij pridobljenih s sledenjem 10 naključnih točk 96
C.3 Množica fazno poravnanih prostorsko-časovnih krivulj 97
C.4 Varianca vsebovana v prvih D lastnih vektorjih 98
C.5 Prvi 4 glavni vektorji (od leve proti desni) 99
C.6 Preslikave množice prostorsko-časovnih krivulj v podprostore 99

List of Tables

5.1 The number of ST-curves extracted from fast walk sequences 44
5.2 Recognition results for varying number of Gaussians 57
5.3 Correlation of learned and observed 1st principal vectors of fast walk sequences 59
5.4 Phase difference matrix of learned and observed 1st principal vectors of fast walk sequences 62
5.5 Summary of recognition results 63

C.1 Povzetek rezultatov razpoznavanja 103

Chapter 1

Introduction

1.1 Motivation

In the near future it will be possible to cost effectively acquire and store video recordings of human activity over extended time periods. Technological advances will continue to provide ever increasing computational capabilities to process recordings in real time. Efficient methods are sought that extract information about human activity for the purposes of surveillance and intelligent environments that can recognize and react to patterns of human activities.

We envision an autonomous system which is able to learn appearance dynamics of humans from past video recordings and apply the information learned to assist recognition and tracking in new recordings. This is a complex unsolved problem which poses several formidable challenges and requires us to constrain research to topics that can be solved adequately with state of the art methods.

In this work a model is sought that captures spatio-temporal dynamics of human locomotion from a monocular camera view over several iterations of the same motion pattern in the part of spatio-temporal domain covered by human appearance, while trying to abstract the corresponding manifold distribution by means of dimensionality reduction and probabilistic modeling. No prior knowledge about the skeletal structure is assumed, instead we want to create a flexible model that we can use to approximate the structure while learning it from the video sequences. Thus we can also learn appearances of people carrying objects.

We attempt to develop robust algorithms to acquire the parameters of the model. Finally we intend to show the proposed model and algorithms are suitable to assist tracking of people, recognition of individuals and recognition of human activity. We also investi-

gate possibilities for transfer of knowledge about motion patterns in the appearance domain from person to person.

1.2 General background

Visual tracking is a field of computer vision research that investigates techniques of object tracking in time sequences of images. These techniques can be applied to people tracking, gait recognition and activity recognition, which in turn apply to automated video surveillance and intelligent environments that can respond to patterns of human activity.

Human body is a relatively complex articulate object with a multitude of possible configurations that stem from bone, muscle, skin, hair and clothes movement. In addition, it can be viewed from a variety of angles, and, when recorded, in different resolutions and different camera perspectives. Even if we were able to record all interesting configurations, the variations in illumination further add to combinatorial explosion.

The appearance of human body as recorded by a CCD camera is simply a set of rectangular colored picture elements that belong to the observed body and not the background. From the apparent multitude of possible configurations we can conclude that storing and indexing all appearances of even a single person is not practical. Even if we took images of a huge number of configurations, we would still be unable to match images of a person wearing new clothes, new hair style or performing newly learned dance routines. We need a more efficient model that abstracts the motion from some aspects of the appearance while retaining individual characteristics.

Several such models have been developed primarily for the purposes of tracking [47, 63, 103, 49, 98, 9, 29, 69, 43, 100, 71, 51], motion capture [28, 77], motion synthesis [37, 10], and gait analysis [60, 61, 53, 54, 102, 38, 72, 5, 20, 33, 101].

Modeling human motion for the purposes of tracking and recognition requires some additional considerations. Humans often use or carry objects. That requires from us not to assume the skeletal structure in advance. We need our model to either adapt to the structure or to be able to adequately model any likely structure.

Finally, when we attempt to learn models of human motions from *a priori* video recordings, we are typically unable to cover the multitude of individual configurations and activities necessary to model all possible or even likely activities that we may observe in new recordings, where we want to use our prior model for tracking or recognition. This requires us to investigate opportunities for knowledge transfer from person to person, and from controlled laboratory environments to uncontrolled realistic environments. The former would allow us to recognize activities a person has never done, while the latter would

allow us to facilitate tracking in difficult conditions including clutter or noise.

Any models and methods developed must also be robust. They must overcome noise present in any realistic situation and model the variations that are inevitably present when a person repeats a motion pattern.

1.3 Problem statement

The problem can be stated as follows:

Given video sequences of persons performing cyclic locomotion develop a method for learning of a spatial-temporal model of individual motion based on a set of trajectories of continuously tracked points on the surface of the person. Develop methods for matching and classification of acquired models. Investigate possibilities for knowledge transfer among models from different persons.

1.4 Thesis contributions

This thesis makes the following contributions to address issues related to people tracking, locomotion pattern recognition, and activity recognition:

- we propose a novel model for cyclic human locomotion representation based on a set of spatio-temporal trajectories of continuously tracked points on the surface of a person;
- we propose a novel methodology for generalization of cyclic locomotion based on principal component analysis and Gaussian mixture models;
- we propose a novel methodology for locomotion model matching and classification based on maximum *a posteriori* likelihood estimate and linear data adaptation.

1.5 Thesis structure

In the first chapter we introduced the motivation, general background, the problem statement and thesis contributions.

In chapter 2 we overview the literature, summarize related work and modeling of locomotion. We introduce our approach and contrast it to related work.

In chapter 3 we introduce the theoretical background to learning and recognition of a probabilistic spatio-temporal model of human locomotion, constituent mathematical methods and notation.

In chapter 4 we detail the implementation of auxiliary methods that complete our framework.

In chapter 5 we present experimental results. First we evaluate parameters and performance of basic constituent methods. We seek practical confirmation for those assumptions that were asserted without firm theoretical basis. We evaluate the performance of our method for the purposes of identification and activity recognition. We conduct testing in uncontrolled outdoor environment to expose issues that need to be addressed in future work.

In chapter 6 we discuss the experimental results. We investigate possibilities for knowledge transfer with respect to results. We propose possible alternative implementations or extensions and outline directions for our future work.

Appendices include a brief summary of Principal Component Analysis and Expectation-Maximization algorithm for diagonal Gaussian mixture model for self-containment of the thesis.

In Appendix C we provide an extended summary of the thesis in Slovenian language.

Chapter 2

Background and Related work

2.1 Review of literature

The bibliography related to the thesis in a wider sense is extensive, however most of the prior works relate to the problem statement only in parts.

Several different approaches that relate in some aspect to our work could be classified into:

- structure from motion
- spatio-temporal
- optical flow

An extensive critique of structure from motion approaches is provided in [62].

Several spatio-temporal models have been developed in [84, 60, 61, 99, 38, 100, 48, 46, 35, 40] with extensive bibliography in [70].

Johansson light displays were discussed in [47, 16, 17, 79].

Barron *et al.* [4] overview and quantitatively evaluate a number of optical-flow based techniques, including instances of differential, matching, energy-based and phase-based methods. Additional work is in [36, 65, 32].

Topics concerning trajectories and tracking were discussed in [75, 7, 52, 92, 18, 29, 71, 43, 31, 68, 93, 50, 67].

Cyclic motion detection and representations were developed in [87, 88, 73, 74, 21, 24, 22, 23, 63, 102].

Principal component analysis and related parameterization approaches are found in [42, 82, 9, 83, 98, 13, 19, 26].

Several correspondence and similarity metrics are proposed in [37, 34, 14, 89, 38, 90, 15, 93, 94, 59, 30, 91, 96].

Gaussian mixture modeling, Expectation-Maximization and Bayesian classification algorithms are developed in [27, 86, 69, 72, 2, 39, 76, 44, 85].

Human gait analysis and recognition can be found in [60, 61, 53, 54, 102, 38, 72, 5, 20, 33, 25, 101].

Works relating to action and behaviour recognition are [68, 46, 67, 66, 11].

Numerous other methods and models have been developed in the field of computer vision for view based tracking and identification of human locomotion incorporating a variety of models for representation. The bibliography is extensive, we found the following works also provide a perspective for our problem: [78, 49, 64, 28, 51, 80, 32, 57, 103, 77, 58, 81, 10, 95].

2.2 Related work

In this section we summarize the most related work and emphasize the contributions.

Johansson 1975: Visual motion perception [47]

Cedras and Shah survey motion analysis from moving light displays [16]. They discuss experiments by Johansson [47] and others that show human observers can recognize gestures, gait and sex from trajectories of lights attached to human actors. They extend their survey to the general case of motion based recognition in [17].

We propose to track random tracked points which are analogous to Johansson light displays in informational content.

Tibshirani 1992: Principal curves revisited [82]

Hastie and Stuetzle [42] introduced the term of *principal curves* which describes a smooth curve passing through the middle of a data cloud, and is a generalization of linear principal components. Tibshirani [82] extends the work on principal curves by introducing an alternative definition based on a mixture model. He carries out estimation by an EM algorithm.

We extend their idea further by probabilistically expressing a set of curves by multiple principal curves.

Niyogi and Adelson 1994: Analyzing and Recognizing Walking Figures in XYT [60], Analyzing gait with spatiotemporal surfaces [61]

Niyogi and Adelson were the first to analyze and recognize walking figures in XYT [60] by recognizing spatiotemporal signature of an XT slice from the lower part of the body, fitting a stick figure and recognizing gait by comparing angle signals. In [61] they model walking figures by smooth spatio-temporal surfaces of temporal evolution of front and back edges. They express observed locomotion by a combination of canonical motion and individual deviation surface.

This is seminal work in the field. It is most related to our approach. We extend their work by probabilistically modeling the whole continuously observable surface over space and time.

Deutscher *et al.* 1999: Tracking through singularities and discontinuities by random sampling [29]

They address issues in markerless tracking of human body motion. They analyze applicability of Kalman filters and circumstances under which the Gaussianity assumption can break down. They illustrate failures in kinematic singularity and at joint endstops. They propose particle filtering or Condensation as robust alternative algorithms for tracking in difficult conditions.

This work support our idea of random sampling of trajectories with no filtering.

Heisele 2000: Motion-Based Object Detection and Tracking in Color Image Sequences [43]

Heisele proposed detecting and tracking objects in a color video sequence by analyzing the motion of clusters built by grouping of pixels in a color/position feature space. The second step is a motion-based segmentation, where adjacent clusters with similar trajectories are combined to build object hypotheses.

We take this idea and probabilistically build clusters of similar trajectories of a single person.

Giese and Poggio 2000: Quantification and classification of locomotion patterns by spatio-temporal morphable models [38]

They develop space-time morphable models for representation of classes of complex movements. They approximate new complex movement patterns by linear combinations

of few learned prototypical example patterns. The weights of the linear combination provide a low-dimensional description of the patterns that can be exploited for the classification of the underlying actions, and also for the estimation of continuous parameters that quantify characteristic properties of the movement. They demonstrate the technique for the classification and quantification of properties of locomotion patterns.

This work provides extensive insight to spatio-temporal modeling of locomotion patterns.

Ormoneit *et al.* 2001: Learning and Tracking Cyclic Human Motion [63]

They estimate a statistical model of typical activities from a large set of 3D periodic human motion data (relative motion of 19 joint angles). They describe an automated method for learning periodic human motions from training data using statistical methods for detecting the length of the periods in the data segmenting it into cycles and optimally aligning the cycles. They present a PCA method for building a statistical eigen-model of the motion curves that copes with missing data and enforces smoothness between the beginning and ending of a motion cycle. The learned eigen-curves are used as a prior probability distribution in a Bayesian tracking framework. Tracking in monocular image sequences is performed using a particle filtering technique.

This work is similar to our approach, but we develop a more general marker-less method.

Vlachos *et al.* 2002: Robust Similarity Measures for Mobile Object Trajectories [93]

They investigate techniques for similarity analysis of spatio-temporal trajectories for mobile objects under the presence of noise. Such kind of data may contain a great amount of outliers, which degrades the performance of Euclidean and Time Warping Distance. They propose the use of non-metric distance functions based on the Longest Common Subsequence (LCSS), in conjunction with a sigmoidal matching function. They compare these new methods to various Lp Norms and also to Time Warping distance.

This work provides good insight to similarity analysis of spatio-temporal trajectories.

Torresani *et al.* 2003: Learning Non-Rigid 3D Shape from 2D Motion [85]

They develop an algorithm for learning the time-varying shape of a non-rigid 3D object from uncalibrated 2D labeled tracking data. They model shape motion as a rigid component (rotation and translation) combined with a nonrigid deformation. They assume

that the object shape at each time instant is drawn from a Gaussian distribution. Their algorithm simultaneously estimates 3D shape and motion for each time frame, learns the parameters of the Gaussian, and robustly fills-in missing data points. They extend the algorithm to model temporal smoothness in object shape, thus allowing it to handle severe cases of missing data.

This work is similar in spirit to ours, but they use labeled data and try to learn the exact shape, whereas we use unlabeled data to learn trajectories of shape in motion.

2.3 Modeling human locomotion

Several methods have been developed in the field of computer vision for view based tracking and identification of human locomotion ([47, 63, 103, 49, 98, 9, 29, 69, 43, 100, 71, 51, 28, 60, 61, 53, 54, 102, 38, 72, 5, 20, 33, 25, 32, 101] etc.). The approaches can roughly be divided into top-down and bottom-up. The top-down approaches typically assume a spatial model and estimate temporal evolution of configuration parameters. The bottom-up approaches attempt to build a spatial model from a set of primitives and continue with temporal evolution.

Spatial models found in the literature are generally image based [5, 21, 9, 32] or geometry based, either 2D [60, 20, 102] or 3D [64, 103, 77, 29]. Cardboard models [49] put 2D image templates in a geometrical structure. For gait recognition, several authors [60, 61, 53, 25, 102, 20, 101] merely track some predetermined features and do not model the entire view space . There have been several attempts of using image based subspace methods [32, 9] and image statistics [72] to decrease the dimensionality of spatial representation, but these methods tend to lose local descriptive power. Most of the methods are limited in representing objects that deviate from the assumptions about the spatial configuration.

Temporal evolution is typically represented by a time series of parameters or by state transitions [103] usually modeled by hidden Markov models [51]. State based methods can easily represent a multitude of different motions, but they are not especially suited to capture details of motion, because too many states would be required to model all possible configurations and local variation.

2.4 Our approach

A lot of work on human motion analysis was performed on data accumulated from motion capture of markers attached to human actors. Cedras and Shah [16, 17] discuss experiments by Johansson [47] and others showing that people successfully recognize human motion from a very small set of markers even in presence of noise.

There have been few attempts of markerless bottom-up structure-free learning of locomotion. Niyogi *et al.* [60, 61] used spatio-temporal manifold produced by evolution of edges of human silhouette over time, however they neither model motion inside silhouette, nor do they model vertical component of motion explicitly. They introduce the idea of canonical motion, but they provide no means of generalizing and learning such a model probabilistically. Torresani *et al.* [85] attempt to learn moving shape from video, but they use labeled data and try to learn the exact shape, whereas we attempt to use unlabeled data to learn trajectories of an articulate shape in motion.

To the best of our knowledge, there have been no attempts to model human locomotion as a space-time manifold of the whole observable surface. The main advantage of such a model is an ability to probabilistically represent structure and motion in a single framework, with possibility to include local and global variation.

We assume that the appearance of motion of an articulate object can adequately be represented by a set of trajectories of points on the surface on the object, if the number of tracked points is adequate to approximate the moving surfaces and degrees of freedom. We focus on cyclic human locomotion, for which a number of databases have been accumulated and additional physical constraints allow us to use a simple point-tracking algorithm and filter out non-optimal trajectories.

We present a novel method for representation of articulate cyclic motion based on a set of spatio-temporal trajectories of continuously tracked points on the surface of the observed object. We apply the method to visual learning and recognition of human locomotion. We assume no prior information about the distribution of trajectories. The main advantage of the method is that it probabilistically models the motion over both full view space and time. At this point our method only includes continuously trackable surface points, but in principle the model can include any trackable features.

The main contribution of this thesis is a method for learning and recognition of the spatio-temporal distribution of a set of spatio-temporal curves over a number of iterations of cyclic motion. We generalize spatio-temporal trajectories over iterations using principal component analysis. Finally, we approximate the distribution using a mixture of Gaussians. The recognition is implemented with a combination of maximum *a posteriori*

estimate and linear data adaptation.

Chapter 3

Probabilistic spatio-temporal model

In this chapter the constituent methods of our approach are introduced and explained in detail.

3.1 Overview

We can divide the approach into three functional sets:

- Spatio-temporal curve extraction from a video sequence of locomotion
- Methods for learning a probabilistic spatio-temporal model of an extracted set of spatio-temporal curves from a locomotion sequence
- Methods for recognition of an extracted set of spatio-temporal curves based on learned prior models of locomotion sequences

The methods are graphically presented as flowcharts in Figures 3.1, 3.7 and 3.11. The following sections introduce the methods in the above order.

3.2 Spatio-temporal curve extraction

The common prerequisite for both learning and recognition of a video sequence of locomotion is spatio-temporal curve extraction. We aspire to attain a set of connected spatio-temporal curves by tracking of random points on the surface of the observed subject.

We assume that we observe locomotion in a semi-controlled environment, which we define to be an environment with largely static background and constant illumination. The extent to which we need to respect these requirements is defined by the performance of

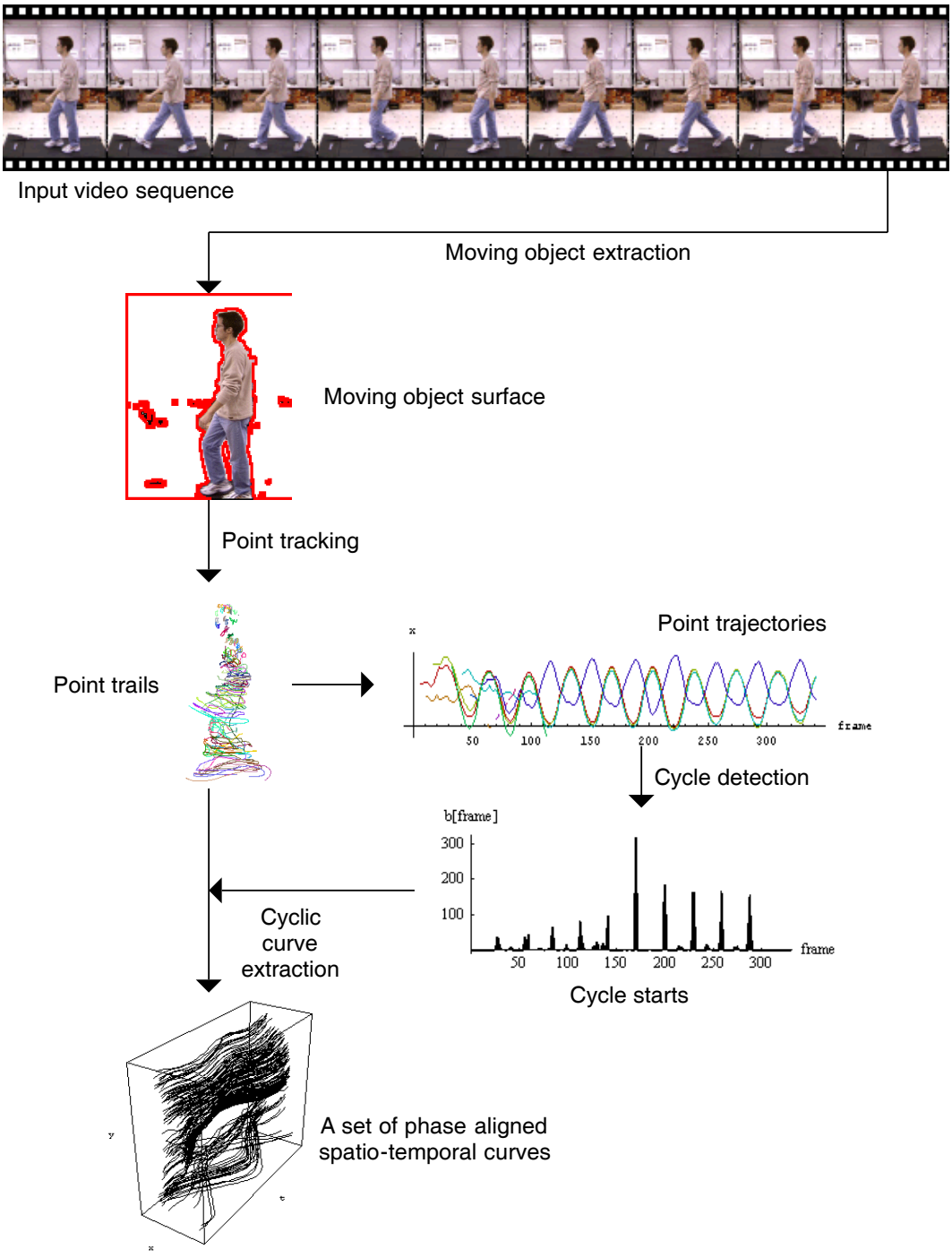


Figure 3.1: Overview of ST-curve extraction from a video sequence of locomotion

segmentation methods. The performance of segmentation is much more critical for the learning phase, because we want to exclude outliers and only learn the significant features that actually belong to the observed subject. The proposed recognition method can treat

outliers probabilistically.

3.2.1 Moving object extraction

The first step of video sequence processing is extraction of a moving object. The idea is to segment the video frame to foreground which likely contains the surface of the moving person, and background which does not contain any interesting features for the task at hand. By using the foreground mask we gain several advantages:

- The foreground mask allows us to contain point-tracking to the surface of the person.
- We do not need to keep point-tracking information for the background points.
- We can also avoid tracking points on the edges of the limbs, which are difficult to track when the limbs cross the torso, by constraining point tracker to track only point inside the foreground mask.
- We can use a simple point tracking algorithm and filter out lost point-tracks that leave the foreground mask.
- Finally, we can filter out some of the background noise that is only temporally classified as foreground by filtering out broken trajectories.

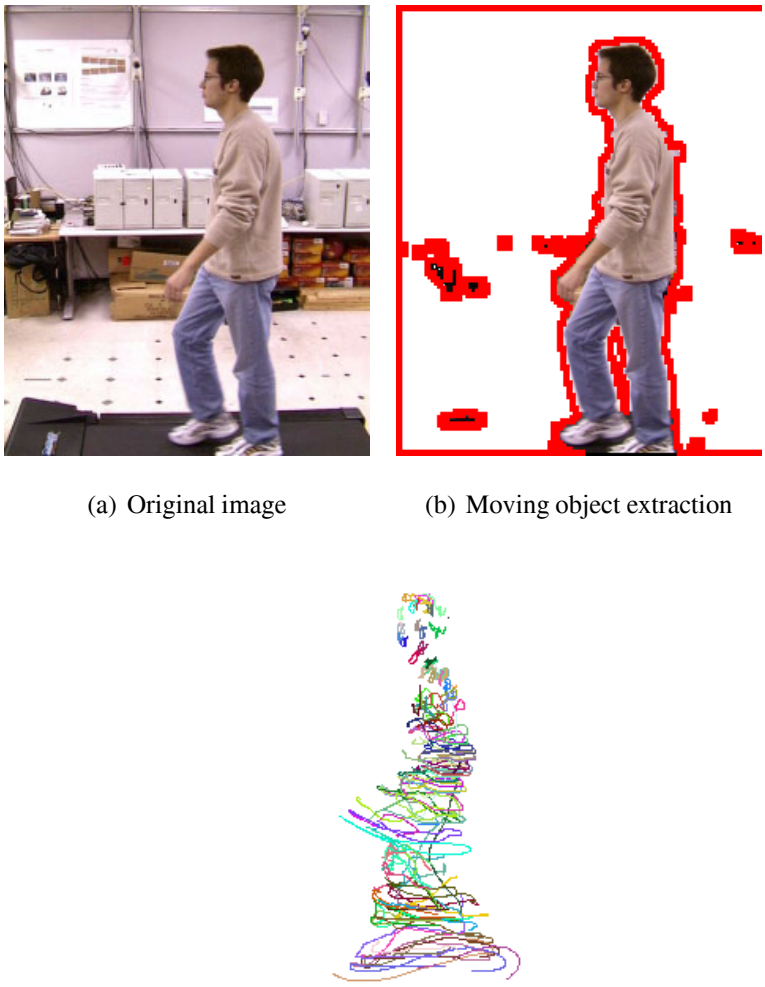
Moving object extraction is a rather mature topic in computer vision and we do not attempt to make any contribution to the theory. We use a standard set of methods for moving object extraction in a semi-controlled environment with a static background, which consists of global illumination normalization, background subtraction, shadow suppression and morphological filtering.

We detail our method in the Implementation chapter.

3.2.2 Point-tracking in a monocular sequence

After the foreground moving object has been segmented a point-tracking algorithm is required to track continuously trackable points on the surface of the observed person.

Instead of attempting to track all points on the surface, we propose random sampling of point trajectories based on optic flow of a small patch around the tracked point. We do not require a specific point-tracking algorithm, we only specify the sampling method should fulfill the following requirements:



(a) Original image (b) Moving object extraction

(c) Point-tracking trails

Figure 3.2: Original image, moving object extraction and point-tracking

- It should assume no prior knowledge of moving object appearance.
- It should attempt to sample the foreground evenly in space and time.
- It should be able to tolerate temporally disappearing parts.
- It should be able to restart random sampling of reappearing parts.
- The density of sampling should be controllable.

We require the following output for each point-trajectory time series:

- coordinates $\boldsymbol{c}(t) = (c_x(t), c_y(t))$ in view-space and

- presence flag $q(t)$ which marks whether the point was successfully tracked in time instant t.

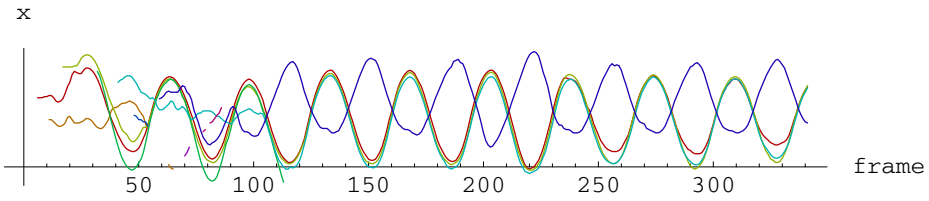


Figure 3.3: Sample trajectories obtained by point-tracking of 10 random points

We present specific details of our algorithm in the Implementation chapter.

3.2.3 Cycle detection and curve extraction

We define a cycle to be a basic repetitive unit, which equals to a time interval containing two steps of a person walking.

We detect cycles in short sequences (around 10 cycles) of locomotion by searching for maxima of autocorrelation of trajectories and voting.

Cycle detection is controlled by parameters C_{min} and C_{max}, defining minimum and maximum cycle. These values can be determined by expressing the minimum and maximum expected cycle size respectively with the number of frames at the frame rate of the video sequence. The center of the sequence is defined as:

$$f_{\frac{1}{2}} = \frac{\text{sequence size}}{2} \ . \tag{3.1}$$

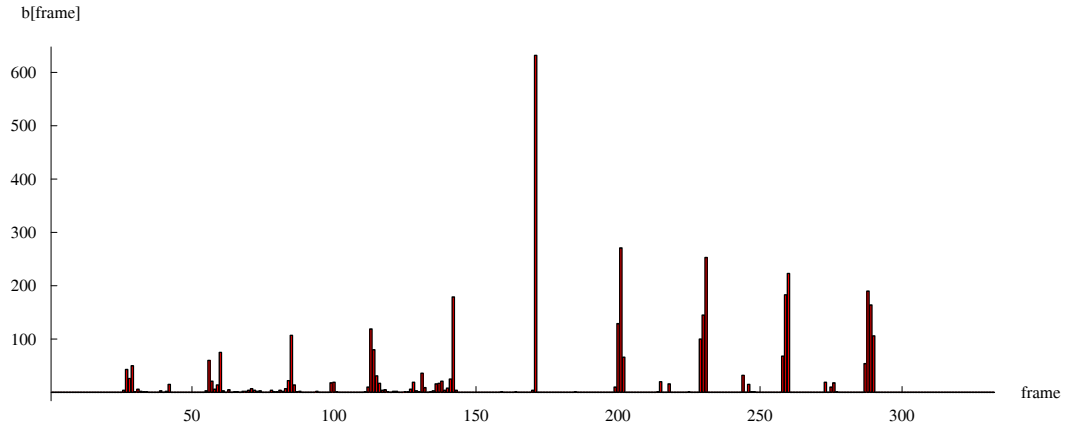
For each trajectory the window of size C_{max} in the center of the sequence is chosen as a reference and autocorrelated against all other windows on the trajectory, where the trajectory is connected ($q(t) = true$) throughout both windows. If the local maximum of autocorrelation in the frame f is also a maximum in interval $[f - C_{min}, f + C_{min}]$, a vote of 1 is added to the voting accumulation vector element $\boldsymbol{b}_f$ (Figure 3.4(a)).

Voting vector $\boldsymbol{b}$ is consequently smoothed by the kernel $(1, 2, 1)$ (see Figure 3.4(b)). Local maxima f, where

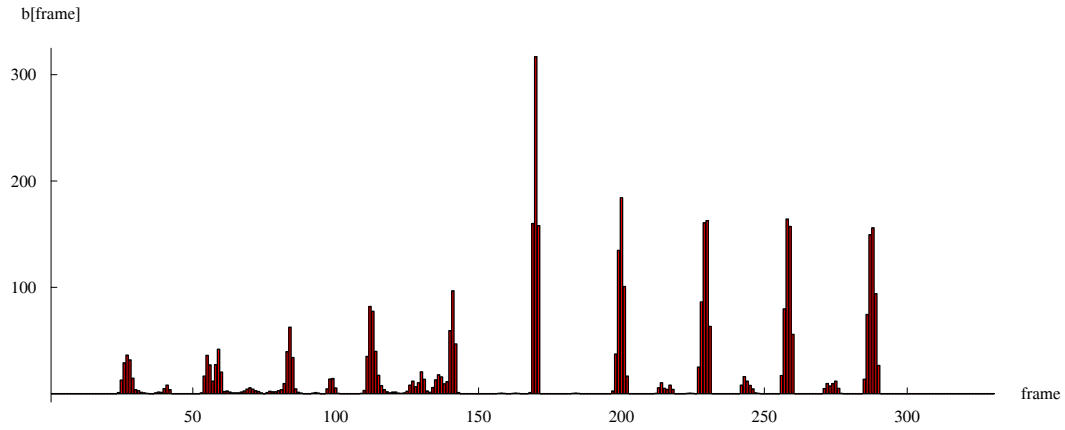
$$\boldsymbol{b}_f = \max_{i=f-C_{min}}^{f+C_{min}} \boldsymbol{b}_i \ , \tag{3.2}$$

define cycle start candidates. Subsequent candidates are subtracted to produce a list of possible cycle lengths. The median cycle size is chosen as the final cycle length estimate C. Starting frames of the candidates that fall within the intervals

$$f_{\frac{1}{2}} + k * C \pm \frac{C}{4} \ ; \ k == -\frac{f_{\frac{1}{2}}}{C} ... \frac{f_{\frac{1}{2}}}{C} \tag{3.3}$$



(a) $\boldsymbol{b}$ without smoothing



(b) $\boldsymbol{b}$ smoothed by kernel (1,2,1)/4

Figure 3.4: Voting accumulation vector $\boldsymbol{b}$ for cycle start candidates

are chosen as final cycle starts to filter out unlikely candidates that appear as noise at half-cycle offsets.

Only the connected parts of trajectories starting from the detected cycle starts are extracted. The result is a set of phase aligned nearly cyclical spatio-temporal curves (see Figure 3.5). At this point each ST-curve is represented by a time series of coordinates from a starting frame f:

$$\boldsymbol{\xi} = [\boldsymbol{c}(f), ..., \boldsymbol{c}(f + C)] \ . \tag{3.4}$$

We compute the difference from the cyclic form by subtracting the endpoints:

$$\boldsymbol{d}_C = \boldsymbol{\xi}_{C+1} - \boldsymbol{\xi}_1 \ . \tag{3.5}$$

We linearly stitch the curves to make them cyclic by updating the time series with a

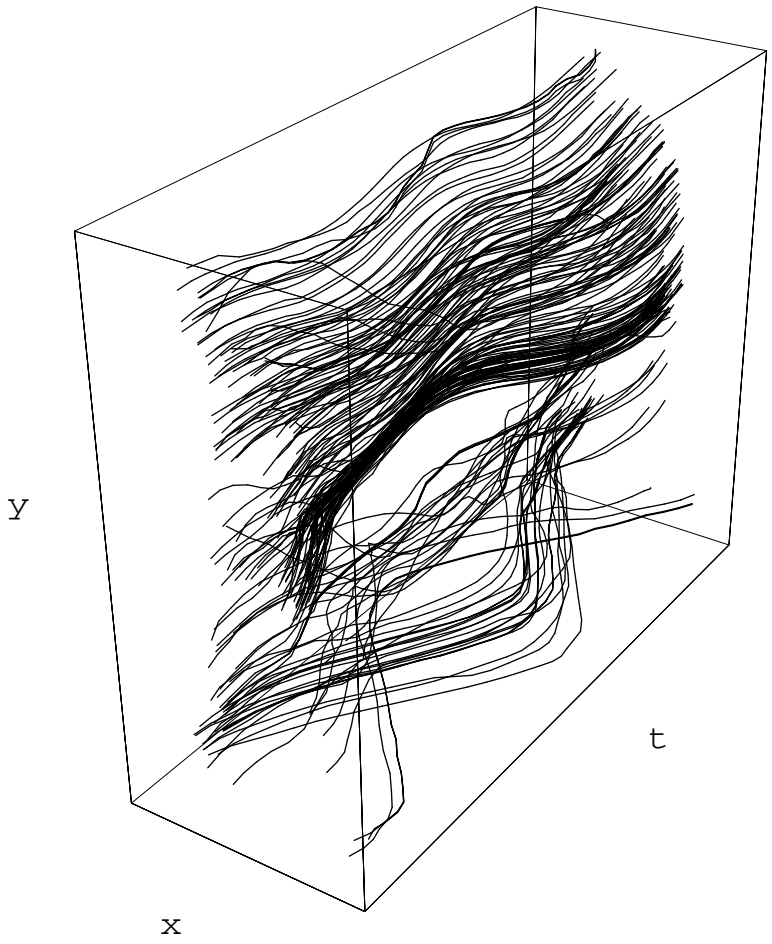


Figure 3.5: A set of phase aligned spatio-temporal curves

linear interpolation of the difference:

$$\boldsymbol{\xi'}_t = \boldsymbol{\xi}_t - \frac{t - \frac{C}{2}}{C} \cdot \boldsymbol{d}_C \; ; \; t = 1...C + 1 \; . \tag{3.6}$$

We treat the updated time series as a piece-wise linear function $\xi(t), t \in [0...C]$. Then, we interpolate the time series to a common size L and subtract the centroid of the curve from the point coordinates:

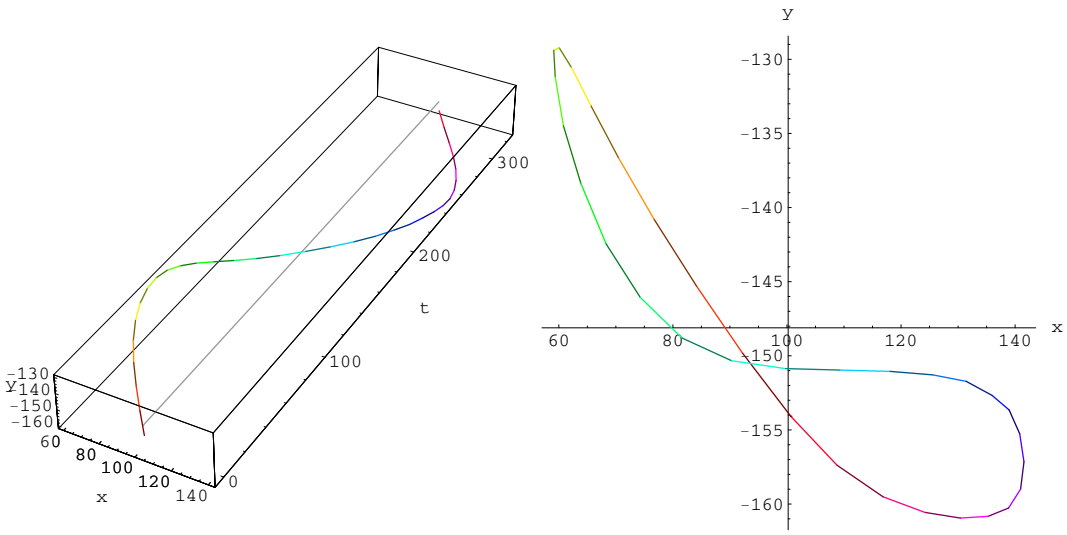
$$\boldsymbol{\xi''}_t = \xi(\frac{(t-1) * C}{L}) \; ; \; t = 1...L \tag{3.7}$$

$$\boldsymbol{o} = \sum_{t=1}^{L} \boldsymbol{\xi''}_t \tag{3.8}$$

$$\boldsymbol{\xi'''}_t = \boldsymbol{\xi''}_t - \boldsymbol{o} \; ; \; t = 1...L \tag{3.9}$$

The final ST-curve representation consists of curve centroid $\boldsymbol{o} = (o_x, o_y)$ and centroid-subtracted curve shape vector $\boldsymbol{x} = \{x_1, y_1, ..., x_L, y_L\}$, where $(x_t, y_t) = \boldsymbol{\xi'''}_t$. All the curves are processed in the same way and the final result is a set of cyclic phase-aligned spatio-temporal curves in centroid/shape representation.

Figure 3.6 illustrates a spatio-temporal curve in 3D and 2D graphs. 3D graph 3.6(a) shows a spatio-temporal curve and its centroid. 2D graph 3.6(b) shows the same spatio-temporal curve in the coordinate system of its centroid. Temporal offsets in both graphs are denoted by hue. The same 2D representation is used throughout this thesis to illustrate single spatio-temporal curves.



(a) 3D curve graph, temporal offset t is scaled for visibility, curve centroid $\boldsymbol{o}$ is grey

(b) 2D curve graph, temporal offset is denoted by hue, curve points are relative to centroid coordinate system

Figure 3.6: 3D and 2D graphs of a spatio-temporal curve

3.3 Learning of a probabilistic spatio-temporal model

In the following sections we describe learning of a probabilistic spatio-temporal model of a set of ST-curves. The learning procedure is divided in PCA decomposition of the curve vectors and subsequent modeling of curve distribution in a subspace by a mixture of Gaussians. Overview of the learning steps is presented in Figure 3.7.

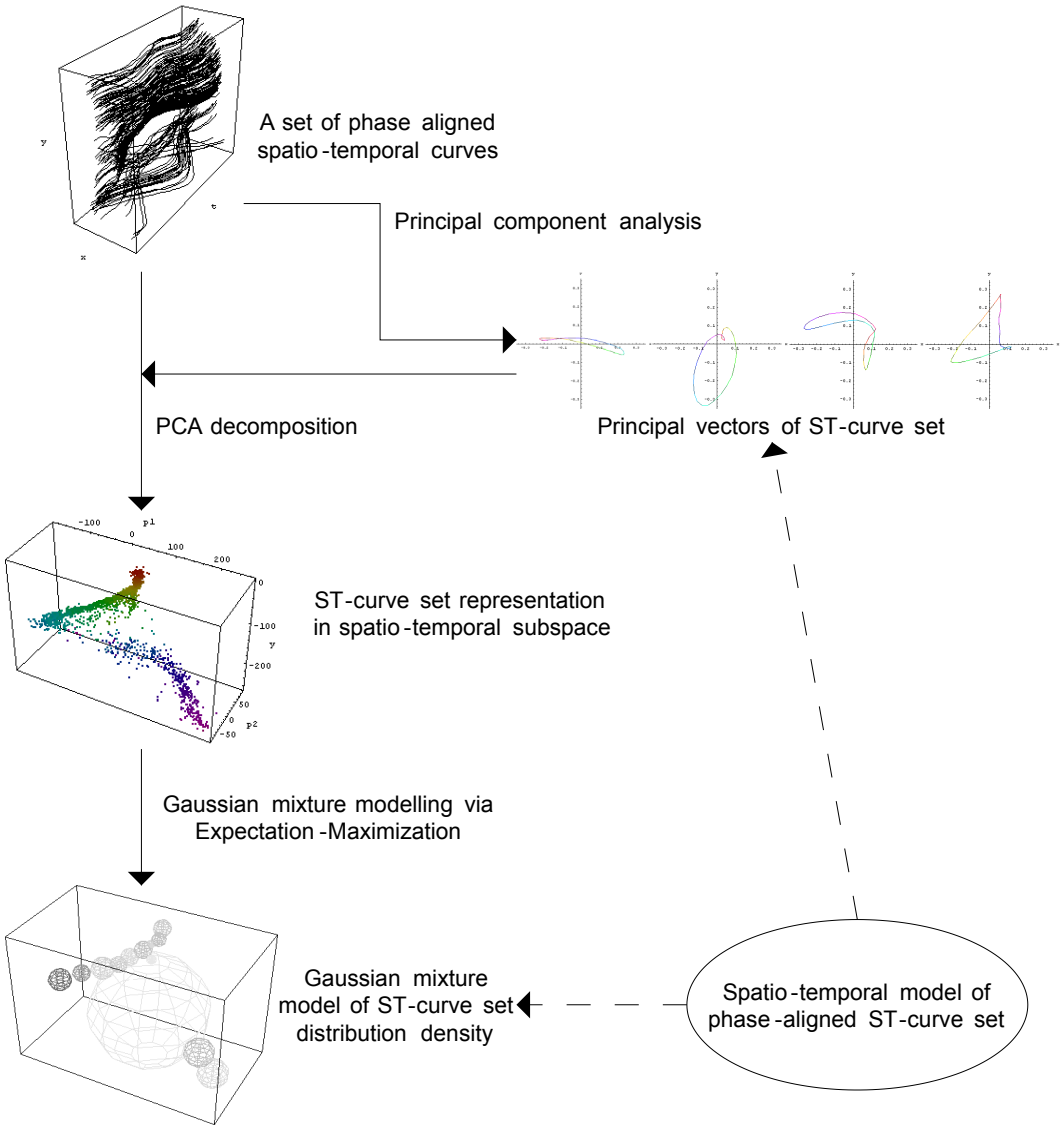


Figure 3.7: Overview of learning of a spatio-temporal model of locomotion

3.4 Curve set in a PCA subspace

Let $\boldsymbol{X}$ be the data matrix with N curve shape vectors $\boldsymbol{x}_n$ of length D ($D = 2 \times L$) in ordered columns. We perform PCA decomposition according to Anderson [3].

$$\boldsymbol{\mu} = [\mu_1, ..., \mu_D]^T = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \tag{3.10}$$

$$\hat{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{\mu} 1_{1 \times N} \tag{3.11}$$

$$\boldsymbol{C} = \frac{1}{N} \hat{\boldsymbol{X}} \hat{\boldsymbol{X}}^T \tag{3.12}$$

By performing eigenvalue decomposition of the covariance matrix $\boldsymbol{C}$ we diagonalize it

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{3.13}$$

in such a way that the orthonormal matrix $\boldsymbol{U}$ contains eigenvectors $[\boldsymbol{u}_1, ..., \boldsymbol{u}_D]$ in its columns and diagonal matrix $\boldsymbol{\Lambda}$ contains eigenvalues λ_i on its diagonal, and the eigenvalues and corresponding eigenvectors are arranged in descending order of the eigenvalues. Thus, the most variability of the set of curves is contained in the first few eigenvectors, also called the principal vectors. We use matrix $\boldsymbol{U}$ to remap curve shape vectors $\hat{\boldsymbol{X}}$ on the principal axes:

$$\boldsymbol{P} = \boldsymbol{U}^T\hat{\boldsymbol{X}} \tag{3.14}$$

The properties of the transform guarantee us that by reducing the representation of the curve vector to the first few principal components we minimize the reconstruction error in terms of mean square error between original and reconstructed vectors.

For a subspace representation of a ST-curve, we choose to use a compositum of 2-dimensional view space to represent curve centroid and D-dimensional PCA subspace to represent its spatio-temporal variation (curve shape). The ST-curve representation in the combined subspace becomes $\boldsymbol{r} \in \mathbb{R}^{2+D}$:

$$\boldsymbol{r} = [o_x, o_y, p_1, ..., p_D] \,, \tag{3.15}$$

where o_x, o_y are the coordinates of curve centroid in the view space, and $p_1, ..., p_D$ are the principal components of the curve shape, as computed in (3.14). Figure illustrates variance contained in the first D eigenvectors. Given the typical variance of a set of ST-curves we expect to reduce the number of principal components p_i significantly ($D = 4...8$) and still retain very accurate representation and reconstruction.

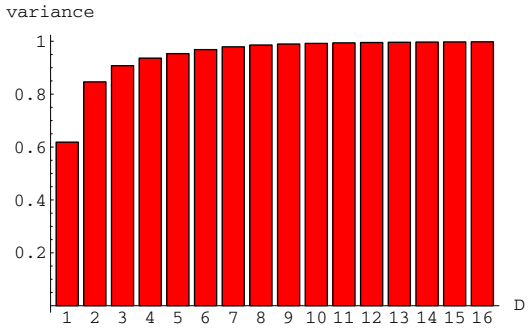
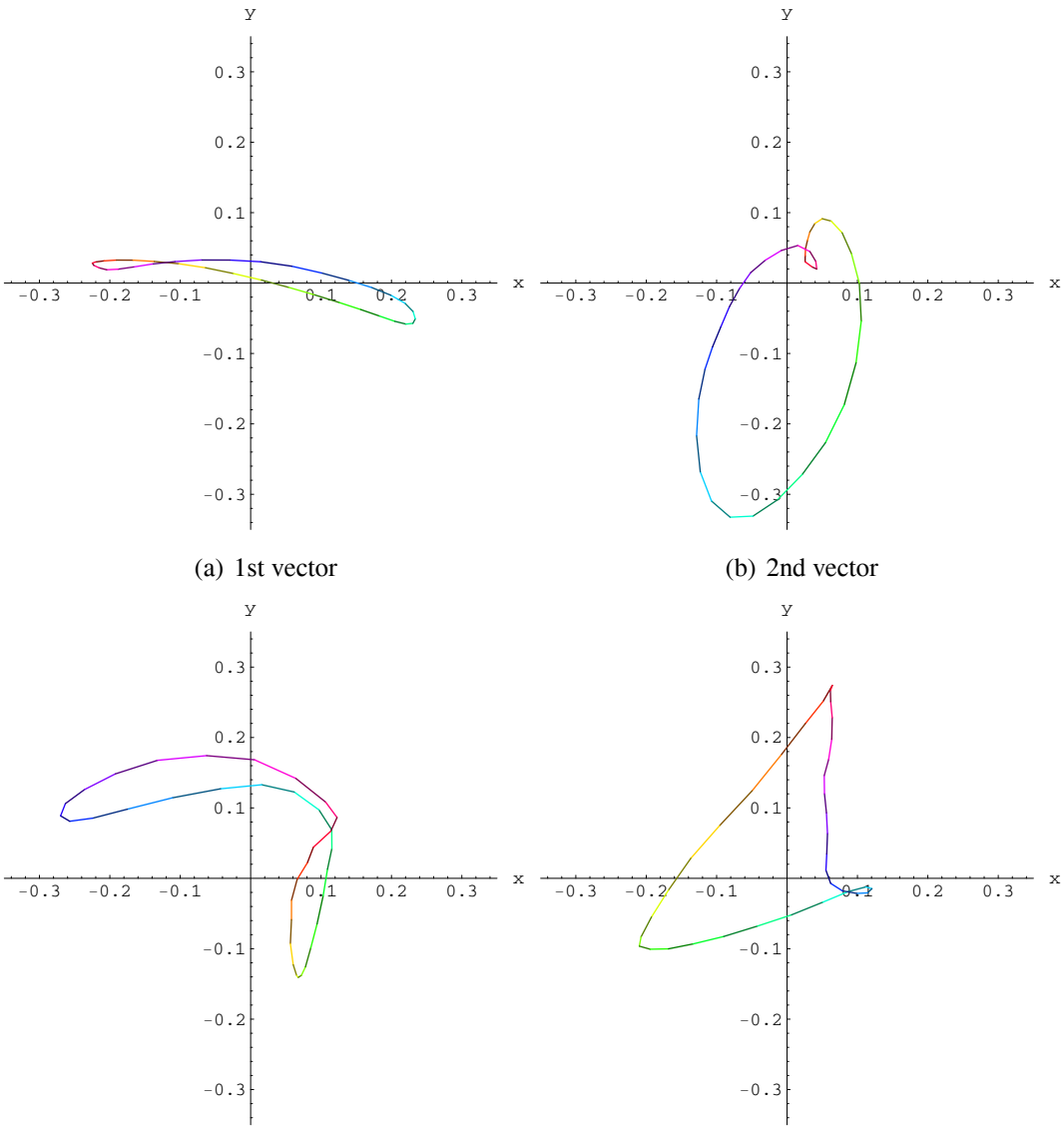


Figure 3.8: Variance contained in the first D eigenvectors



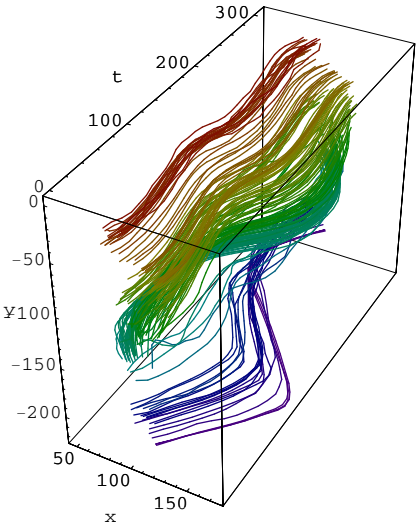
(a) 1st vector (b) 2nd vector

(c) 3rd vector (d) 4th vector

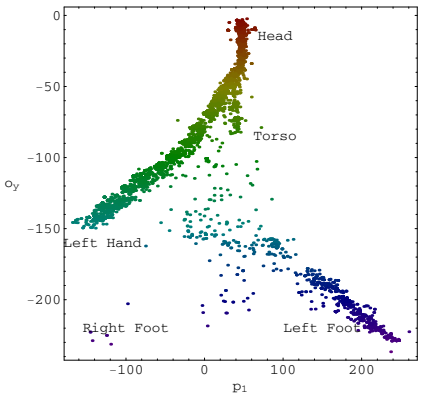
Figure 3.9: The first 4 principal vectors (hue encodes temporal offset)

We analyzed the diagrams of the principal vectors. In all of the cases of a side view the first principal vector is nearly cyclical and contains significant oscillation along the direction of locomotion (see Figure 3.9). This feature is further used for phase alignment of curves.

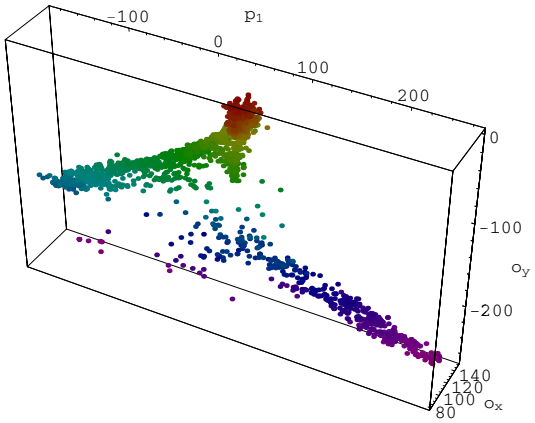
The diagram of the curve set in the subspace of the vertical component o_y and the first principal component p_1 (Figure 3.10(b)) turns out to be reminiscent of a human stick figure, often used as a model for tracking or gait analysis. The difference here is that horizontal component represents the first principal vector of spatio-temporal curve instead



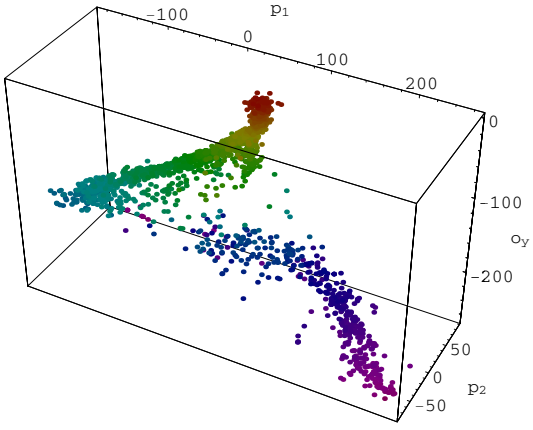
(a) Original set of ST-curves (every 10th curve)



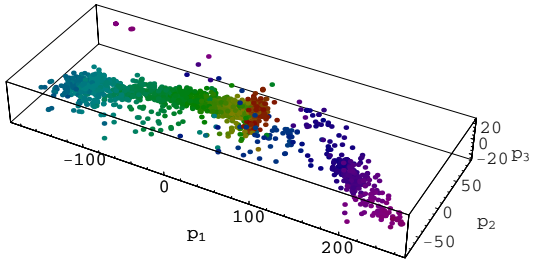
(b) o_y and p_1



(c) o_x, o_y and p_1



(d) o_y, p_1 and p_2



(e) p_1, p_2 and p_3

Figure 3.10: 2D and 3D subspace mappings of ST-curve set (hue encodes o_y)

of spatial offset, and some outliers are included due to errors in point tracking. Two limbs are missing due to occlusions and no attempt of our point tracking algorithm to handle non-connected trajectories. Three-dimensional mappings with largest variance, Figure 3.10(c) and Figure 3.10(d) are very similar, however mappings to curve shape subspace as in Figure 3.10(e) do not preserve much spatial separation except for the increasing amplitude of extremities.

3.5 Subspace modeling of a spatio-temporal curve set

The number of curves extracted from a video sequence is potentially very large. We seek a more compact representation that adequately represents the curve set. By adequately representing the curve set we mean more specifically:

- the model covers the whole representation space
- it preserves distinctive properties of the curve distribution
- the abstraction can intuitively be understood where data is discarded

Thus, before choosing a representation, we need to look at the properties of the spatio-temporal curve sets that we want to model.

3.5.1 Properties of a spatio-temporal curve set of locomotion

There are several properties of the curve sets of locomotion that need to be addressed, some of which can actually help us in the temporal abstraction part.

- The curves are accumulated from several (3–10) cycles of locomotion, and we do not distinguish them by cycle.
- The curves of nearby points on the tracked object naturally form clusters. The curves from the same cycle thus result in a cluster that primarily captures spatial distribution. The curves of the same points from different cycles are more spread in the prior space, but we expect a single point to form Gaussian distribution over a large number of locomotion cycles.
- Visually, the curve set in a subspace projection slightly resembles a stick figure (see Figure 3.10). The extremities with maximal amplitude and different curve shapes are spread out, but as we follow points toward the trunk of the body, the projected

points close in on the center of the subspace. The central clusters basically represent points with low amplitude on the trunk and head, shape vector is close to 0, and only the vertical component varies (representing the altitude of the curves, or effectively the o_y component of the points).

- There are two types of outliers, which can result from the point tracking phase:

  1. The first type is caused by the curves of the tracked points that are not on the target objects. We can filter some of these outliers if they are sufficiently far from the majority of the curves, or if they have very small amplitude.
  2. The second type is caused by the curves of the tracked points on the object, that got mistracked by the point tracking algorithm. This can happen due to low texture, noise or temporary occlusions, which cause the point tracker to jump to another similar point. We can filter some of these outliers by checking that the beginning and end of the curve are sufficiently close to form a cyclic curve. Alternatively, we can speculate that observing the same object will result in similar tracker errors and then these curves can be used as regular data.

- The distribution is clearly multimodal and with a varying density. There are three major reasons for distribution density variation:

  1. Trajectories of tracked points vary over cycles. The variation of different points of the body is not uniform, partially due to the nature of motion and partially due to non-cyclic moves (change of gaze, arm gestures, change of cycle, stability corrections etc.).
  2. Some regions are prone to losing tracked points, due to texture noise (caused primarily by wrinkling clothes) or partial occlusions.
  3. The texture affects the stability and density of tracked points. Low texture increases the probability of mistracking, which decreases the density of tracked points. High texture results in more stable tracking with a close to uniform density.

3.5.2 Alternative spatio-temporal curve set representations

We explored several variants for ST-curve set representation.

1. The simplest approach is to keep the set as-is and find an appropriate similarity metric.

   There are several problems with this approach:

   - The number of data points can be potentially huge, how do we efficiently find neighbors?
   - What is the appropriate similarity metric?
   - How do we account for different density of prior and observation sets and how does it affect similarity metric?
   - How do we efficiently spatially align datasets?
   - How do we handle outliers?

   On the positive side of such an approach would be determinism.

2. Many of the these problems can be alleviated by a form of clustering. We expect the curves of neighboring points on the surface to be similar in space-time and therefore naturally form local clusters, or at least they can be approximated with clusters.

   We evaluated K-means clustering. Deterministic clustering has several immediate advantages:

   - Subsets of data points can be represented by clusters, thus decreasing the complexity of finding neighbors.
   - Local density of the distribution can be estimated based on the distribution of data points in the cluster.
   - Global probability of a cluster can be estimated based on the number of data point that are closest to the cluster according to some metric.

   However, if we want to represent the data set probabilistically, there are global solutions better than estimation of each cluster distribution locally. Local cluster solutions increase the unwanted effect of outliers.

3. Gaussian mixture models are a general approach to probabilistic clustering. Mathematically, the distribution density of a data set is represented as a mixture of Gaussians.

   The important features of Gaussian mixture models for our case are:

- The model probabilistically represents the whole probability space, and approximates the global distribution probability density function.
- Each Gaussian cluster abstracts the spatio-temporal distribution of the curves in the locality, both in local and global terms. The local parameters are the parameters of the Gaussian distribution and the global parameters are the position and the prior probability of the cluster.
- Similarity function can be expressed as *a posteriori* likelihood, given a prior model.
- Learning and recognition can deal with outliers probabilistically.

The only major problem with Gaussian mixture modeling is its probabilistic nature. We need to build several models and find a method to choose the best one. We also need to determine the appropriate number of Gaussians to adequately represent and not overfit the ST-curve set.

3.5.3 Gaussian mixture model of curve distribution

We approximate the density of the ST-curve distribution in a subspace with a mixture of Gaussians Θ. Let $\boldsymbol{r}$ represent a curve vector in a subspace as defined in (3.15), $\boldsymbol{\mu}$ a Gaussian mean, and $\boldsymbol{\Sigma}$ a covariance matrix. The probability density function of a single Gaussian is given as:

$$\mathcal{N}(\boldsymbol{r};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{|\boldsymbol{r}|}{2}}\sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{r}-\boldsymbol{\mu})} \tag{3.16}$$

The probability of a curve vector $\boldsymbol{r}$ given a mixture of K Gaussians Θ can then be expressed as:

$$p(\boldsymbol{r}|\Theta) = \sum_{i=1}^{K} w_i * \mathcal{N}(\boldsymbol{r};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)\ . \tag{3.17}$$

K is the number of Gaussians and w_i is the weight of Gaussian i with $\sum_{i=1}^{K} w_i = 1$ and $\forall i : w_i \geq 0$.

$\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ denote respective Gaussian means and covariances. We use diagonal $\boldsymbol{\Sigma}_i$ with $\sigma_1 ... \sigma_K$ on diagonal.

$$\boldsymbol{\Sigma}_i = \sigma_i \cdot \boldsymbol{I} \tag{3.18}$$

Gaussian mixture models are universal approximators of distribution densities, given a large enough number of Gaussians. Even diagonal Gaussian mixture models are universal

approximators. The term *diagonal* refers to the diagonal form of correlation matrix $\mathbf{\Sigma}$. Full rank models are very complex to calculate with, because the number of parameters is the square of the number of dimensions. Gaussian mixture models can be trained by maximum likelihood using an efficient algorithm: Expectation-Maximization [27].

The number of Gaussian means

The typical approaches to choose the number of Gaussian means are to preset K according to predetermined criteria, or to determine the optimal number of Gaussians by some statistical or information theoretic criteria. Extensive theoretical and practical insight is provided by McLachlan in [56].

First, K must be large enough to approximate the feature distributions. For our case, practical test show that small K decreases recognition rate. The model becomes less sensitive to where exactly the ST-curves were sampled. Given the shape of the distribution density, some datapoints may not be covered corresponding to their probability. It can also happen that cluster centers do not appear inside data distribution – they may appear outside and then increase posterior likelihood of outliers.

Large K introduces the tendency of overfitting. The distribution density model may become too compact. Overfitting also increases probability of modeling outliers and less likely datapoints (instead of filtering them). Too large K may also introduce problems in the training process, as the amount of data becomes insufficient for a statistical model of many parameters, and the computational cost becomes excessive.

We argue there exists an appropriate K, which sets the number of clusterlets along distribution for a good coverage of all probable model points analogously to centering clusters in *bones* and the local distribution representing the motion of the *surface* around the bones. We can then choose K empirically by observing animations of Gaussians for varying K and subjectively estimating the quality of reconstruction and the distribution of clusterlets along the limbs.

Initial Gaussian mixture model

We initialize the means of Gaussians by setting them to a random subset of data vectors. We initialize variance to a random fraction of observed data interval. Other often used approaches are to initialize by K-means or some other form of clustering.

Expectation-Maximization

We use an iterative Expectation-Maximization [27, 6] procedure to update GMM:

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{N} \boldsymbol{r}_i P(j|\boldsymbol{r}_i, \Theta^s)}{\sum_{i=1}^{N} P(j|\boldsymbol{r}_i, \Theta^s)} \tag{3.19}$$

$$(\hat{\sigma}_j)^2 = \frac{\sum_{i=1}^{N} (\boldsymbol{r}_i - \hat{\boldsymbol{\mu}}_j)^2 P(j|\boldsymbol{r}_i, \Theta^s)}{\sum_{i=1}^{N} P(j|\boldsymbol{r}_i, \Theta^s)} \tag{3.20}$$

$$\hat{w}_j = P(j|\Theta) = \frac{1}{N} \sum_{i=1}^{N} P(j|\boldsymbol{r}_i, \Theta^s) \tag{3.21}$$

Expectation-Maximization procedure locally optimizes the parameters of Θ. Each step is guaranteed to monotonically increase likelihood. We can stop the optimization after a predetermined number of iterations or when the increase in likelihood is below a certain threshold.

However, Expectation-Maximization does not find the global maximum (except for some very specific cases of distributions). It converges towards one of the local maxima. Therefore, we run Expectation-Maximization several times from different random initializations and choose the result that maximizes log expectation:

$$\log p(\boldsymbol{r}_1 ... \boldsymbol{r}_N | \Theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} w_k * \mathcal{N}(\boldsymbol{r}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \ . \tag{3.22}$$

3.5.4 A probabilistic spatio-temporal model of locomotion

The final model of observed motion thus consists of:

- principal vectors $\boldsymbol{u}_1 ... \boldsymbol{u}_D$ and mean $\boldsymbol{\mu}$ modeling spatio-temporal variation of trajectories and
- a set of GMM parameters $\{w_i, \boldsymbol{\mu}_i, \sigma_i\}$, $i = 1...K$ modeling distribution density of trajectories in combined 2-dimensional view space and D-dimensional spatio-temporal curve subspace.

3.6 Recognition

The steps for recognition are the same as for learning up to PCA decomposition, *i. e.* moving object extraction, point tracking, cycle detection and ST-curve extraction. From that point we continue with spatio-temporal alignment and maximum *a posteriori* estimate for classification. Overview of the recognition steps is presented in Figure 3.11.

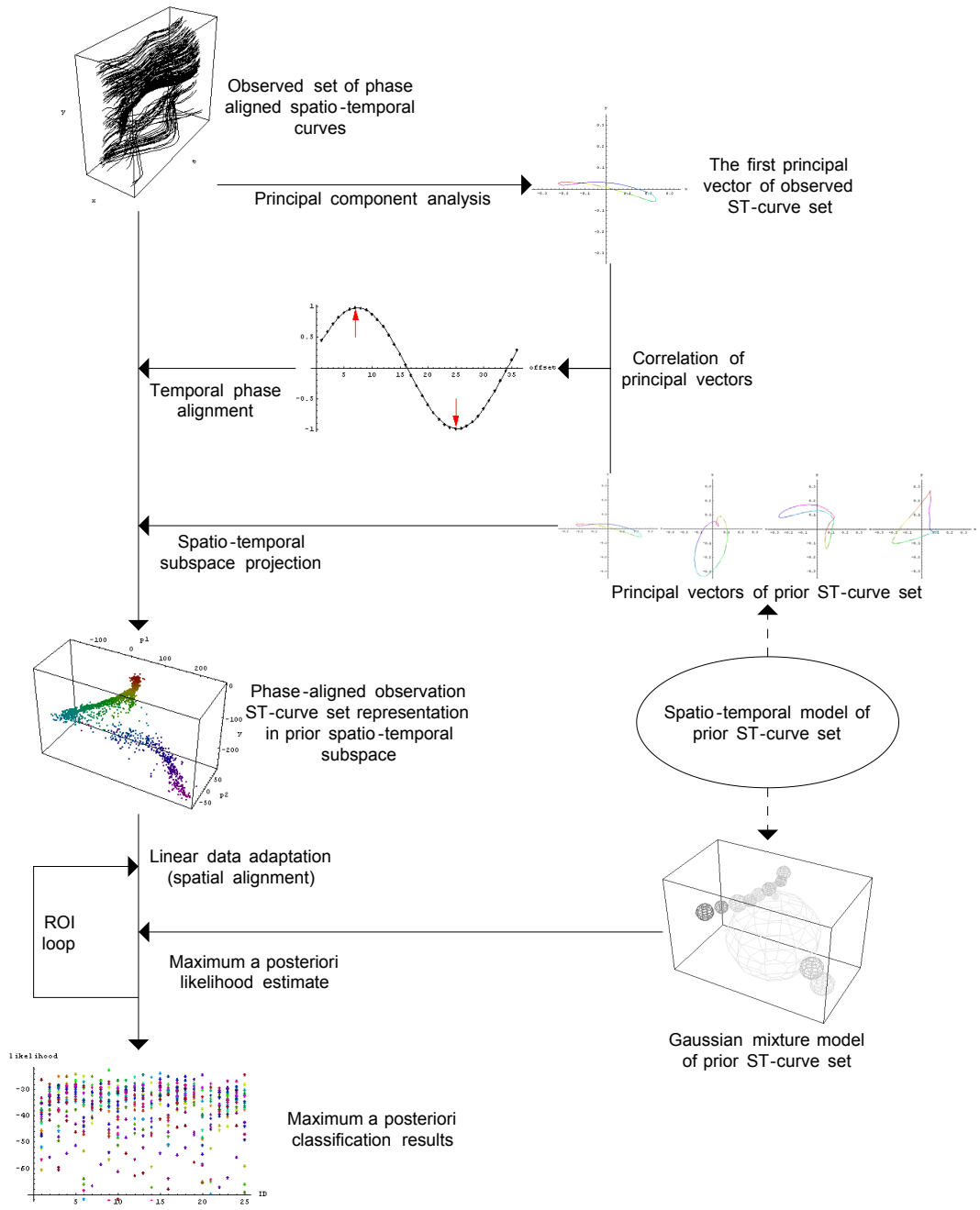


Figure 3.11: Overview of recognition of a locomotion pattern given a prior spatio-temporal model

3.6.1 Spatio-temporal alignment

Given a set of new observation trajectories, we first align them temporally by computing the first principal vector of the observed set of ST-curves and choose the phase that max-

imizes correlation with principal vector of the prior model. We account for both original and negated principal vectors, giving us two possible results for the phase offset, because we cannot assume the orientation of the principal vector.

We assume that an approximate spatial alignment can be attained by other methods, therefore we can use exhaustive search in a relatively small area of interest, which we perform by linear adaptation of data vectors.

3.6.2 Linear data adaptation

The general idea of linear data adaptation is to perform a linear transformation on observed data before classification in order to align it with prior (learning) subspace data distribution.

The acquired spatio-temporal curve set of an observed sequence of locomotion is not necessarily aligned in view space and in scale, even when viewed from the same or very similar angle. In order to seek correspondence of prior and observation data it is necessary to align prior and observation coordinate systems. We assume the view is close enough to orthogonal that we can adjust the difference by a linear transform. This does not mean that we require orthogonal view to use our methods, we merely require the difference between prior and observation sequence views to be close enough that a linear transform is a reasonable approximation. Mathematically we need to perform linear data adaptation on observation data before we proceed with the classification steps. Also, by linearly adapting the data, we can test our methods for scale and shift invariance without requiring additional video sequences.

The general form of linear data adaptation given an observation vector $\boldsymbol{y}$ is:

$$\boldsymbol{y}' = \boldsymbol{A} \cdot \boldsymbol{y} + \boldsymbol{b} \tag{3.23}$$

There are known solutions in the literature for the general problem of linear data adaptation with diagonal Gaussian mixture models. Boulis [12] and Afify [1] use an Expectation-Maximization solution for speaker identification. However, we cannot reasonably assume that the solution of general form of linear data adaptation is the correct result for our problem, because we do not expect independent rescaling of principal axes to give the correct classification results. It would merely maximize likelihood of a given adaptive transformation, we would still need to check if the transformation is viable. Also, we have no advanced knowledge of the distribution, so a simple maximization of likelihood may end up in a mode that is unviable with an unknown distance from the closest naturally viable correspondent configuration.

Therefore, to develop and test our recognition method, we will use the special case of affine translation and orthogonal scaling:

$$\boldsymbol{A} = s \cdot \boldsymbol{I} = \begin{bmatrix} s & & & & 0 \\ & s & & & \\ & & s & & \\ & & & . & \\ & & & & . \\ 0 & & & & s \end{bmatrix} \tag{3.24}$$

$$\boldsymbol{b} = [o_x, o_y, 0, ..., 0] \tag{3.25}$$

The parameter s is the scaling factor and the parameters (o_x, o_y) represent the offset correction from the prior in view space. This linear transformation works, because in our ST-curve representation only the first two components are offset dependent (they represent the offset of the curve in the spatio-temporal subspace which is represented in view-space coordinates), and the principal components of curves are offset independent (their centroid is subtracted from the representation prior to PCA decomposition).

Linear data adaptation allows us to use the same Gaussian mixture model on scaled and translated models, and a theoretical extension to include rotation $\boldsymbol{R}$ is straightforward.

$$\boldsymbol{y}' = \boldsymbol{R} \cdot (\boldsymbol{A} \cdot \boldsymbol{y} + \boldsymbol{b}) \tag{3.26}$$

By using linear data adaptation we generalize our model to a scalable model, acquiring a significant advantage over miscellaneous non-scalable models from the literature that require multi-resolution or hierarchical extensions when observable data is scalable.

The minimal requirements for linear data adaptation applicability are:

- maximum *a posteriori* estimate of self-classification corresponds to a shift of $(0, 0)$ and scale of 1,
- correct self-classification when the observation set is shifted or scaled.

In this thesis we do not concern ourselves with a mathematically optimal solution for the data adaptation problem, instead we want to test if the linear adaptation solutions fit our expectations.

3.6.3 Classification

For the classification step we compute maximum *a posteriori* likelihood of the observed ST-curve set given a prior model.

We start with the spatio-temporally aligned trajectories from the previous step and recompute a new data matrix of observation vectors $\boldsymbol{Y} = [\boldsymbol{y}_1 ... \boldsymbol{y}_N]$ in the remapped space with principal axes from each prior model c.

We want to find the model c_i that maximizes posterior probability $p(c_i|\boldsymbol{Y})$ given a set of curves $\boldsymbol{Y}$. We use the Bayes rule:

$$\arg\max_i p(c_i|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|c_i)p(c_i)}{p(\boldsymbol{Y})} \ . \tag{3.27}$$

For our case $p(\boldsymbol{Y})$ is the same for all models, since the point tracking method is the same for all models and independent of them. Moreover, each observation $\boldsymbol{y}_j$ within $\boldsymbol{Y}$ is independent of other observations.

We do not include probability of particular model c_i in our framework, we define the prior probability $p(c_i)$ of all models to be equal.

Thus, assuming equally likely models c_i and noting that $p(\boldsymbol{Y})$ is the same for all models, the classification simplifies to

$$\arg\max_i p(\boldsymbol{Y}|c_i) \ . \tag{3.28}$$

Assuming independence between observations, the final recognition rule is simplified to:

$$\arg\max_i \prod_{n=0}^{N} p(\boldsymbol{y}_n|c_i) = \arg\max_i \sum_{n=0}^{N} \log p(\boldsymbol{y}_n|c_i) \ . \tag{3.29}$$

The model c_i that maximizes (3.29) is selected as the most likely candidate.

3.6.4 Application to recognition and tracking

Recognition is implemented by computing (3.29) for each class while linearly adapting data to cover the area of interest. Note that computing $p(\boldsymbol{y}_n|c_i)$ requires $\boldsymbol{y}_n$ be remapped to the principal subspace of c_i for recognition, practically requiring spatio-temporal remapping of data matrix $\boldsymbol{Y}$ and computation of $\sum_{n=0}^{N} \log p(\boldsymbol{y}_n|c_i)$ for each class separately.

For the purposes of tracking we may merely be interested in the posterior probability of a single class. Thus tracking can be implemented by searching for maximum *a posteriori* likelihood in the area of interest, which is expressed by linear data adaptation parameters. The presented approach also requires that the translation be subtracted from the trajectories before proceeding with curve extraction. There are several methods in the literature (*e. g.* [60]) which achieve that.

Chapter 4

Implementation

In this chapter the developed set of auxiliary algorithms is introduced and explained in detail. These algorithms do not represent a significant scientific contribution to the thesis, but are nevertheless an important part of the framework required for the application of proposed methods and a prerequisite for the interpretation of results.

All the algorithms are implemented in a GUI based Windows application programmed in C++ (see Figure 4.1).

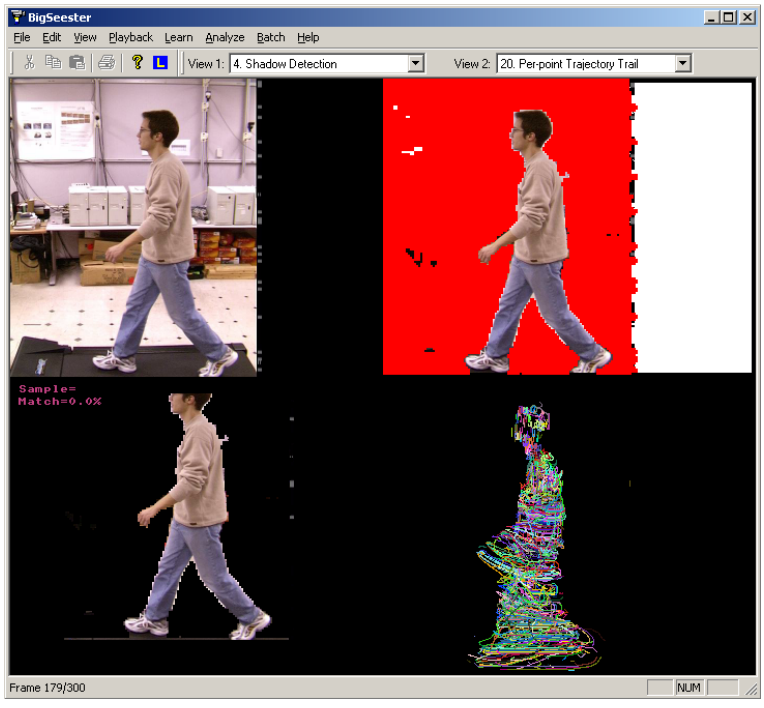


Figure 4.1: Screenshot of the application during point tracking

4.1 Video input preprocessing

The input is a video sequence in a video format readable by Microsoft DirectShow filters or a sequence of JPEG images. We have successfully processed videos in JPEG, MPEG, M-JPEG and DV-AVI formats. Each frame is decompressed to a linear buffer and downscaled to half PAL resolution which is 360×288. Downscaling step was implemented to avoid interlacing effects and JPEG quantization noise.

4.2 Moving object extraction

This section presents the algorithm used to extract a moving object from a video sequence using a static background model. The algorithm consists of:

- global luminance normalization,
- background subtraction,
- shadow suppression and
- morphological filtering.

The output is a foreground mask marking the pixels that belong to a moving object.

4.2.1 Global luminance normalization

Global luminance normalization is required because common input video sequences contain significant luminance variations over time due to mainly two factors:

- impact of light source frequency and
- exposition variation over time.

We assume that the background is largely static and only a small part of the image contains moving objects. We estimate the magnitude of global illumination by computing a median of the pixel luminance values. This value is computed for each incoming frame and for the reference background image. The current frame is normalized against the reference background image.

4.2.2 Background subtraction

A Gaussian model of each pixel component in YUV space is used as a static background model. Let $\boldsymbol{p}_k(n)$ represent the value of pixel element k in frame n. The background model is initialized using a video sequence of the static background.

$$\bar{\boldsymbol{p}}_k = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{p}_k(n) \tag{4.1}$$

$$\sigma_k = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{p}_k(n) - \bar{\boldsymbol{p}}_k)^2} \tag{4.2}$$

Background subtraction is performed by a hysteresis filter with the following thresholds:

$$threshold_{k,fg} = 3.6\sigma_k \tag{4.3}$$

$$threshold_{k,bg} = 1.8\sigma_k \tag{4.4}$$

A pixel is classified as foreground when it exceeds $threshold_{k,fg}$ in any component, and reclassified as background when all components are below their $threshold_{k,bg}$.

4.2.3 Shadow suppression

Shadow detection is implemented by comparing block luminance and difference against background model.

A square block of 4×4 pixels is luminance normalized and compared against normalized block in the background model to detect shadows. Let $\boldsymbol{l}$ represent the vector of pixel luminance components, and $\boldsymbol{p}$ represent the vector of all pixel components, and let $\boldsymbol{p}_{fg}$ and $\boldsymbol{p}_{bg}$ mark current and background pixel vectors respectively.

$$\bar{\boldsymbol{l}} = \frac{1}{4\times 4}\sum_{i=1}^{4\times 4}\boldsymbol{l}_i \tag{4.5}$$

The average of absolute differences per normalized component is computed for classification.

$$\Delta = \frac{1}{4\times 4}\sum_{i=1}^{4\times 4\times 3}\left|\frac{\boldsymbol{p}_{fg,i}}{\bar{\boldsymbol{l}}_{fg}} - \frac{\boldsymbol{p}_{bg,i}}{\bar{\boldsymbol{l}}_{bg}}\right| \tag{4.6}$$

Blocks that conform to luminance test $0.25\bar{\boldsymbol{l}}_{bg} < \bar{\boldsymbol{l}}_{fg} < 1.2\bar{\boldsymbol{l}}_{bg}$ and difference test $\Delta < 16.0$ are treated as shadow and reclassified as background for the following steps.

The thresholds were determined empirically based on tests on video sequences from a consumer DV camera.

4.2.4 Morphological filtering

The noisy points are filtered by a simple morphological operator. Foreground pixels with fewer than $\frac{1}{4}$ foreground pixels in a 5×5 neighborhood are reclassified as background. The remaining foreground pixels are used as a foreground **object mask** for point tracking.

4.3 Point-tracking in a monocular sequence

This section presents an algorithm for tracking of random points in a monocular video sequence of human locomotion.

4.3.1 A point-model for point tracking

Since we assume no prior knowledge of moving object appearance, we use a small 5×5 pixel patch of texture as a current point neighborhood model. A point model for tracking is defined with the following variables:

- coordinates $\boldsymbol{c}(t) = (c_x, c_y)$,
- velocity $\boldsymbol{v}(t) = (v_x, v_y)$,
- presence flag $q(t) \in \{true, false\}$, and
- texture vector $\boldsymbol{patch}(t)$.

4.3.2 Initialization

Tracking is initialized in the second frame by choosing a predetermined number of random points R. If a point is inside the moving object mask, it is added to the list of tracked points, its texture patch is copied, its velocity is initialized by optical flow relative to the previous frame, and the point is marked as present.

4.3.3 Tracking

In each succeeding frame all the points in the list are tracked. We assume that point velocity changes slowly, therefore a point is predicted to move with constant velocity:

$$\boldsymbol{c}'(t+1) = \boldsymbol{c}(t) + \boldsymbol{v}(t) \tag{4.7}$$

$$\boldsymbol{v}'(t+1) = \boldsymbol{v}(t) \tag{4.8}$$

Patch matching around the predicted position is used with a penalty function to correct displacement and update velocity. Only patches inside the moving object mask are compared constraining trajectories to the surface of the moving object.

Let $\boldsymbol{c}'(t+1) = ({c'}_x, {c'}_y)$ represent predicted coordinates and (d_x, d_y) displacement from the predicted coordinates. Let $\boldsymbol{block}(t)$ represent the vector of pixels values around the tracked point. Let imageblock() be a function that extracts a block of pixels from the target coordinates, and Δ a similarity metric for blocks of pixels. The following function is used to optimize displacement:

$$\arg\min_{d_x,d_y} \Delta(\boldsymbol{block}(t) - \text{imageblock}({c'}_x + d_x, {c'}_y + d_y)) + \text{Penalty}(d_x, d_y) \tag{4.9}$$

$$Penalty(d_x, d_y) = S * \sqrt{d_x^2 + d_y^2} \tag{4.10}$$

The penalty function (4.10) penalizes displacement (d_x, d_y) from the predicted position. Scale S is chosen primarily to filter out noise in very dark areas.

The patch similarity metric Δ is a sum of absolute differences. If the sum of patch similarity and penalty function (4.9) is within a predefined threshold, the current point is flagged as tracked in current frame, and coordinates and velocity are updated:

$$\boldsymbol{c}(t+1) = \boldsymbol{c}'(t+1) + (d_x, d_y) \tag{4.11}$$

$$\boldsymbol{v}(t+1) = \boldsymbol{v}'(t+1) - (d_x, d_y) \tag{4.12}$$

$$q(t+1) = true \tag{4.13}$$

The current texture patch is stored for subsequent matching.

If the point is not successfully tracked, the predicted values are kept for successive frames:

$$\boldsymbol{c}(t+1) = \boldsymbol{c}'(t+1) \tag{4.14}$$

$$\boldsymbol{v}(t+1) = \boldsymbol{v}'(t+1) \tag{4.15}$$

$$q(t+1) = false \tag{4.16}$$

Even if the point is not successfully tracked, it is kept on the tracking list.

4.3.4 Adding new random points

After the points in the list have been tracked in a frame, R random points are generated as in the initialization step, but an additional constraint for new points to be added to the tracking list is that there is no other tracked point within 2 points Manhattan distance. Thus the number of simultaneously tracked points has an upper limit while the parameter R defines the probability of retracking temporally lost or reappearing points.

The algorithm continues by alternating tracking and new point generation until the end of the sequence. Figure 4.2 illustrates the number of successfully tracked points throughout 25 test sequences. We expect the number of tracked points to settle at around $\frac{\text{object area in pixels}}{9}$. A typical person from our test sequences covers (see Figure 3.2(b)) around 12000 pixels putting an upper limit on concurrently tracked points to around 1333.

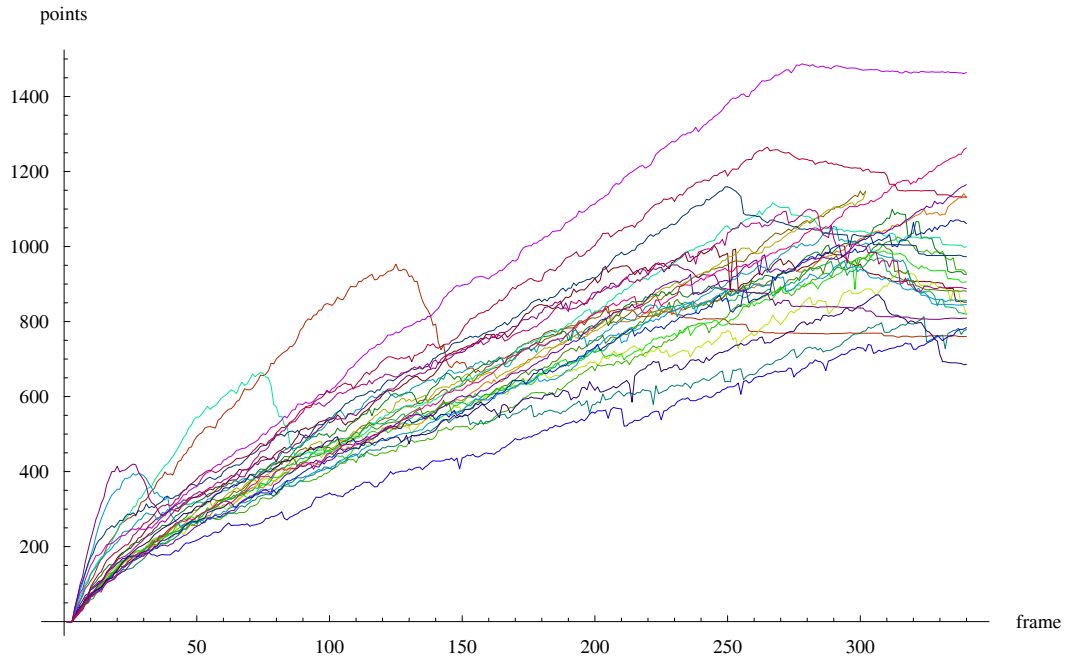


Figure 4.2: Number of successfully tracked points throughout 25 fast walk sequences

4.3.5 Point tracker output

Each tracked point produces a trajectory which is not necessarily connected and sometimes erroneously jumps among parts of object (see Figure 4.3).

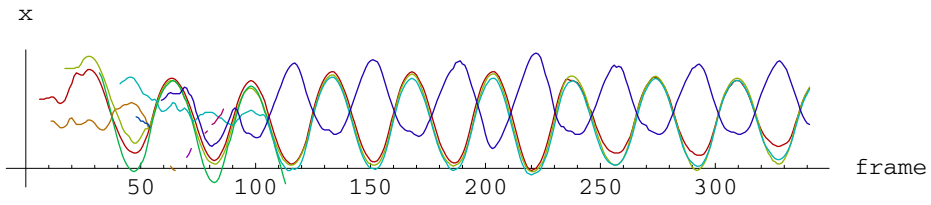


Figure 4.3: Sample trajectories obtained by point-tracking of 10 random points

Chapter 5

Experimental Results

This chapter presents experimental results on CMU MoBo database [41] which was used for the majority of tests of the proposed methods. A test was also done on video sequences recorded with a consumer DV camera to identify issues in a practical outdoor setting.

5.1 CMU MoBo database

We used a subset of CMU MoBo database [41] for experiments. This is a database of short video sequences of 25 people performing 4 types of locomotion (slow walk, fast walk, walking with a ball and incline walk) on a treadmill captured from 6 different angles simultaneously. The sequences contain 300-340 frames recorded at NTSC resolution and framerates.

We concentrated most of our attention on sequences of 25 people captured from the side view, because the trajectories captured from this view exhibit most spatial dynamics potentially useful for recognition. For the same reason, most experiments were done on

(a) Fast walk (b) Slow walk (c) Incline walk (d) Ball walk (e) Slow walk 45°

Figure 5.1: Modes and angles of experimental MoBo locomotion sequences

Person ID	Cycle length	Cycles detected	ST-curves in 1st half	ST-curves in 2nd half	ST-curves per cycle in 1st half	ST-curves per cycle in 2nd half
1	28	10	2253	4054	237 333 448 564 671	759 805 751 839 900
2	31	9	2173	3758	433 634 502 604	682 785 767 766 758
3	33	8	1472	2967	198 296 434 544	611 696 775 885
4	32	7	1685	2316	255 375 489 566	655 767 894
5	32	7	1603	2332	249 341 452 561	673 785 874
6	35	8	1244	2412	171 249 373 451	515 556 623 718
7	29	9	1592	3580	273 370 431 518	592 654 735 780 819
8	30	9	1329	3499	213 299 373 444	520 632 691 787 869
9	36	8	1222	2643	149 255 361 457	542 626 730 745
10	34	7	1533	2107	231 334 438 530	618 715 774
11	32	9	1437	3573	227 317 412 481	578 696 741 757 801
12	33	8	1667	3461	301 333 465 568	678 825 939 1019
13	33	8	1268	2091	190 284 384 410	421 474 555 641
14	35	8	1401	2949	157 310 436 498	596 703 783 867
15	31	9	1487	3609	229 325 430 503	592 647 735 798 837
16	33	8	1856	3628	285 405 532 634	743 900 1015 970
17	31	8	1212	3743	317 405 490	598 679 743 836 887
18	31	9	1036	2556	174 218 289 355	368 455 512 579 642
19	32	8	1289	2191	251 298 362 378	471 449 596 675
20	39	7	1156	3135	248 391 517	636 739 825 935
21	35	8	2169	4865	285 447 640 797	966 1136 1316 1447
22	29	10	2211	3956	206 321 446 587 651	713 798 820 819 806
23	30	9	1989	4096	319 459 565 646	714 779 888 858 857
24	31	8	1593	3253	247 333 444 569	689 728 866 970
25	35	8	1859	4046	258 403 522 676	819 977 1115 1135

Table 5.1: The number of ST-curves extracted from fast walk sequences

fast walk sequences. Figure 5.1 illustrates modes and angles of the chosen sequences.

We performed curve extraction with $R = 100$ random points inspected in each frame, point velocity change limited to $d_x, d_y \in [-4, 4]$, displacement penalty factor $S = 1$, and patch similarity threshold set to $5 \times 5 \times 3 \times 32$. We noticed that estimated cycle size varied from 28 to 39 frames. We used piece-wise linear interpolation to make all curves of equal size which we defined to be $L = 32$ points.

We divided the sequences in half. Table 5.1 illustrates typical numbers of extracted

curves. We already presented the number of tracked points throughout the same sequences in Figure 4.2. The number of curves extracted from the first half is typically smaller for two reasons. There are relatively few tracked points spawned in the beginning, whereas stable points get tracked till the end. Also, some sequences had a lot of noise caused by illumination changes relative to the background model in the beginning of the sequence. Therefore we chose the second half to be used for training and the first half to be used for testing.

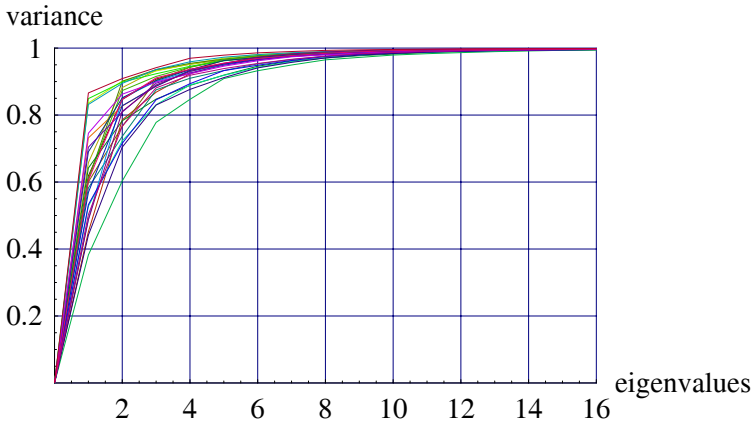
5.2 Learning

The following sections describe experiments related to learning of the proposed spatio-temporal model.

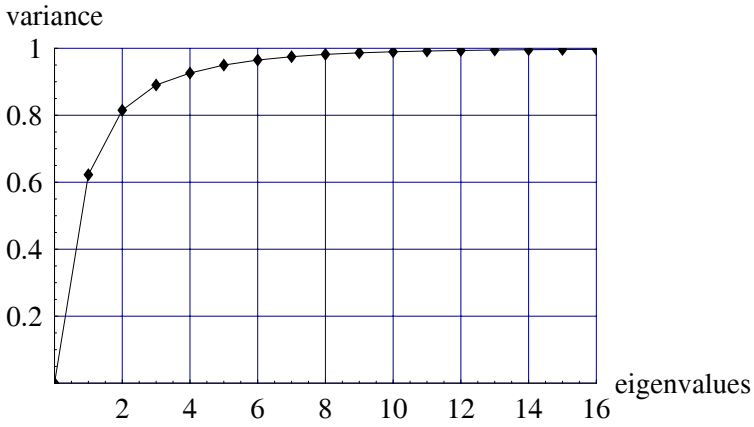
5.2.1 PCA decomposition and reconstruction

Using the described methods we processed the sequences and analyzed their principal vectors (see Figure 3.9) and the quantity of variance contained by the first few principal vectors (see Figure 3.8). We noticed that the first principal vector contains 38.2%–86.6% of variance, the first 4 vectors contain 84.7%–96.9% of variance and the first 16 vectors contain more than 99.4% of variance. We conclude that a very small number of linear basis vectors is enough to accurately represent shapes of ST-curves of locomotion, and the majority of low variance principal vectors contain noise and minor variations.

We synthesized animations of reconstructed locomotion cycles by animating points along reconstructed curves. Each successfully extracted ST-curve resulted in a single animated point, and all the points were blended in a single animation. We visually inspected the resulting animations, and confirmed that as little as 4 principal components are enough for close to perfect reconstruction. Only a few sequences required more than 4 principal components for good reconstruction, mainly due to noise introduced by tracking errors in dark low-texture areas. The mistracked points show as wildly moving points in the reconstructed sequence, because their trajectory cannot accurately be described in the subspace. However, despite the simplistic point-tracking method, there were relatively few such outliers.



(a) Variance data for 25 fast walk sequences



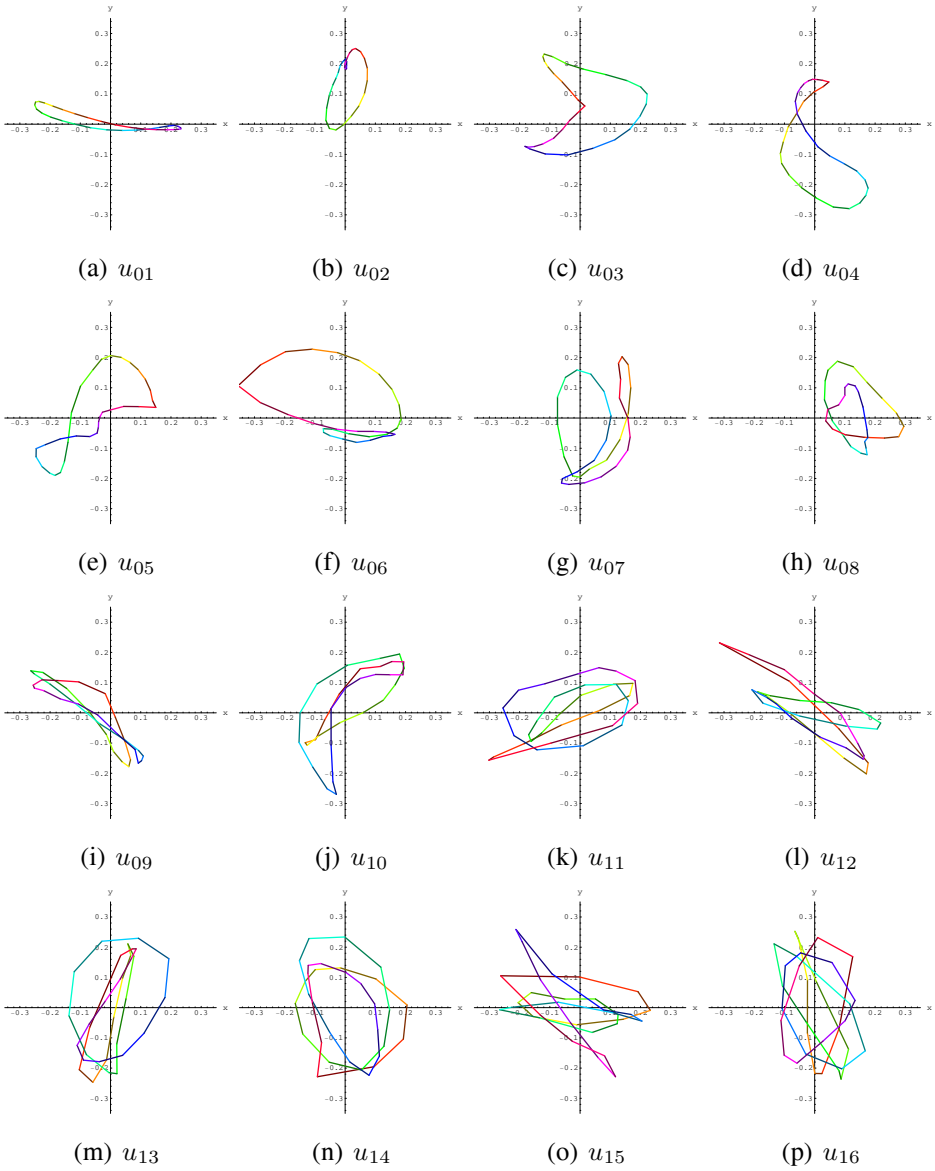
(b) Mean variance of 25 fast walk sequences

Figure 5.2: Cumulative share of variance contained in the principal eigenvalues

5.2.2 Analysis of parametric similarity

We analyzed the diagrams of mean and basis vectors computed by performing PCA on ST-curve sets extracted from the MoBo sequences. We display the vectors as a view-space mapping of the corresponding curve, where the hue encodes temporal offset, or by cyclically animating line segments between the points of the corresponding time-series.

If we look at the principal vectors of a single person performing a mode of locomotion as illustrated in Figure 5.3, we notice that the major principal vectors are rather smooth. As the variance represented by the eigenvector decreases, so does the smoothness. It is obvious that the eigenvectors after the first 8 mostly represent noise, as we could already conclude from the diagrams in Figure 5.2.



(a) u_{01} (b) u_{02} (c) u_{03} (d) u_{04}

(e) u_{05} (f) u_{06} (g) u_{07} (h) u_{08}

(i) u_{09} (j) u_{10} (k) u_{11} (l) u_{12}

(m) u_{13} (n) u_{14} (o) u_{15} (p) u_{16}

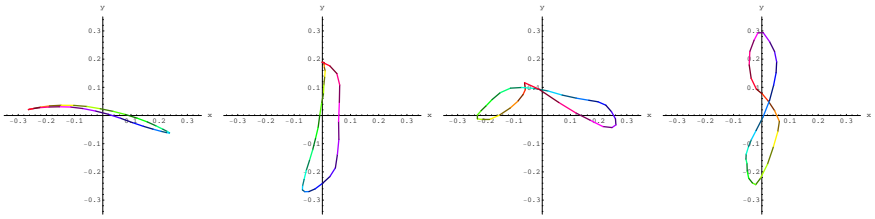
Figure 5.3: The first 16 principal vectors of a fast walk learning sequence (hue encodes temporal offset)

As an illustration of variability between different persons performing the same locomotion mode in the same view, we provide diagrams of the principal vectors of 25 people performing fast walk. The mean is illustrated in Figure 5.5, and the first four principal vectors in Figures 5.6, 5.7, 5.8, 5.9 respectively.

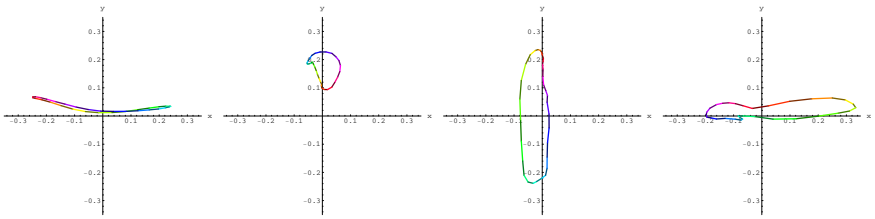
The most striking similarity is that in all of the cases of a side view the first principal vector (see Figure 5.6) is very similar and contains significant oscillation along the direction of locomotion. This feature is further used for phase alignment of curves.

The other principal vectors do not show such resemblances, at least not in an orderly fashion. We can notice a lot of circular shapes and eight-shapes, but that is very subjective. We cannot even say that these curves by themselves represent some familiar biological motion.

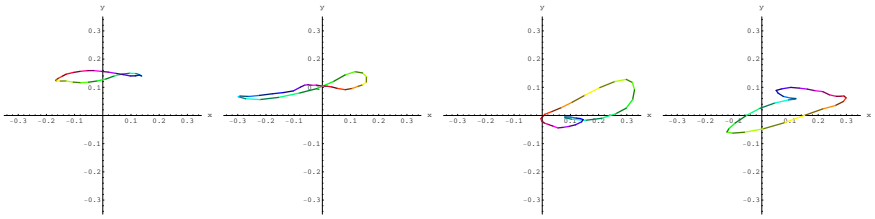
The means are not crucial, because they only cause a linear shift of projections in our subspace, but they do not change the shape of the distribution.



(a) fast walk



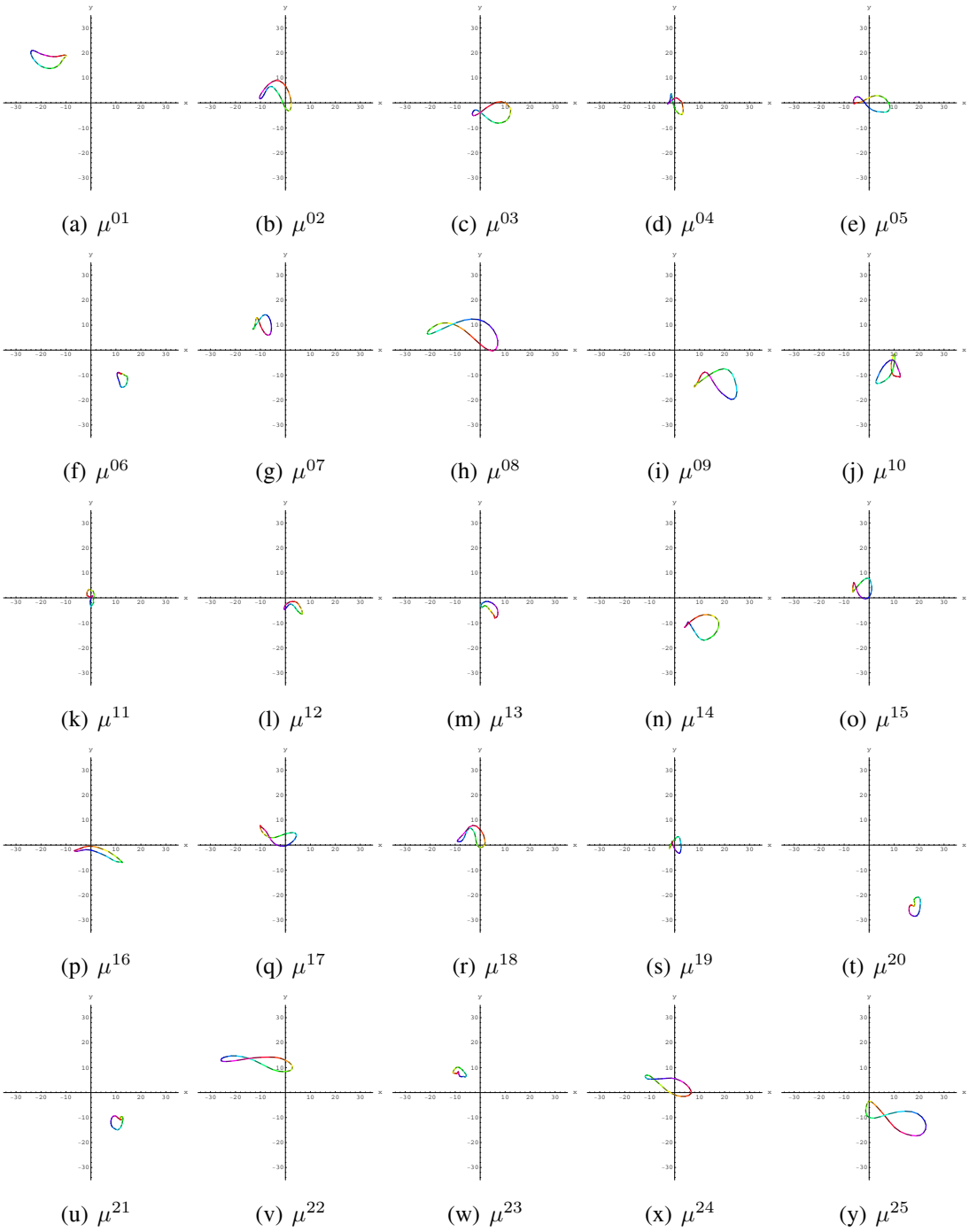
(b) slow walk



(c) ball walk

Figure 5.4: Comparison of the principal vectors for different locomotion modes of a single person (u_1, u_2, u_3, u_4 left to right)

If we look at the differences between principal vectors of a single person performing different locomotion modes (see Figure 5.4), or is being captured from another view, we notice that the curves are quite different, except the first principal vector manages to represent the oscillation along the direction of locomotion (however, this is not the case any more if we analyze the back or front view, where this component is not apparent in the 2D view-space).



(a) μ^{01} (b) μ^{02} (c) μ^{03} (d) μ^{04} (e) μ^{05}

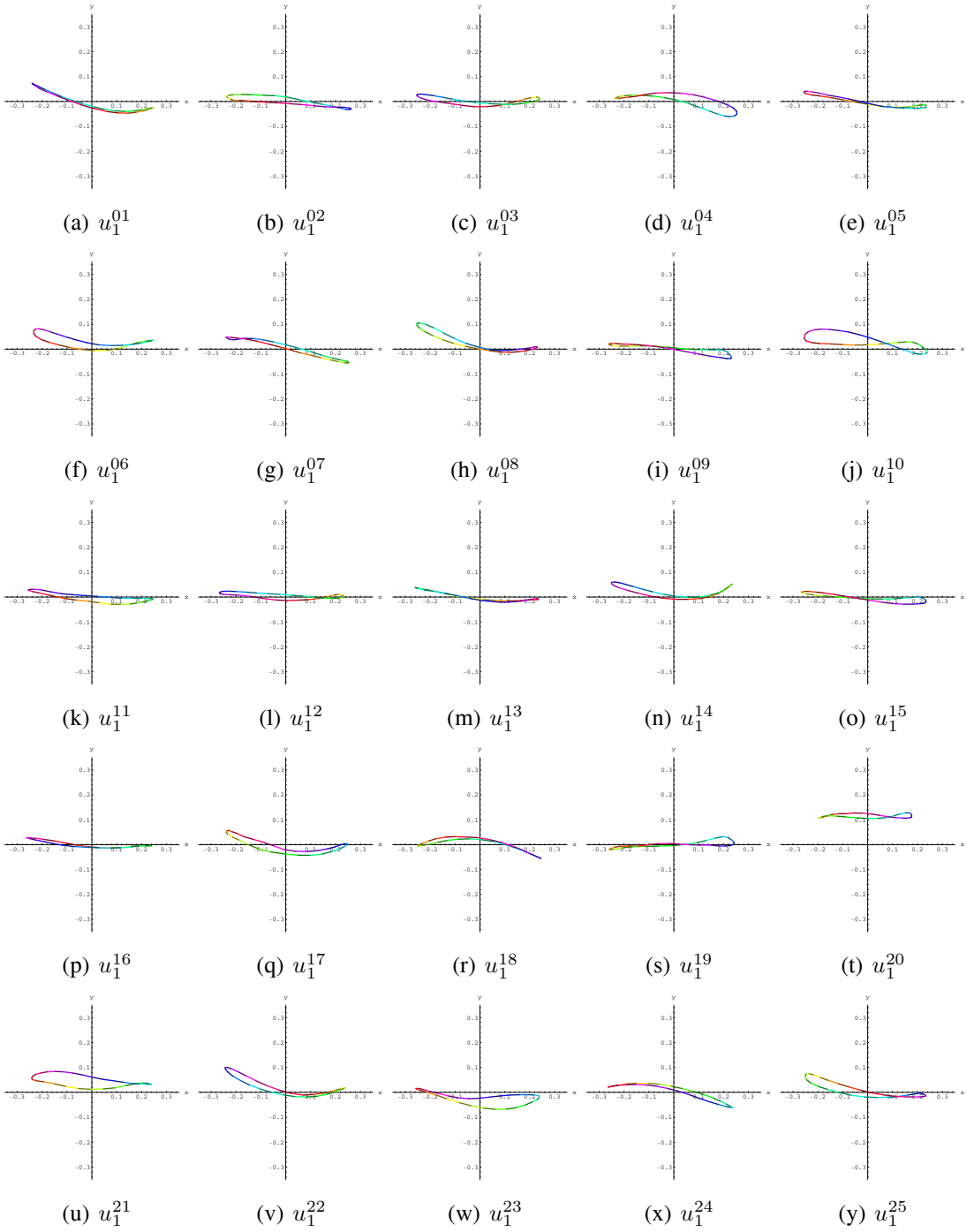
(f) μ^{06} (g) μ^{07} (h) μ^{08} (i) μ^{09} (j) μ^{10}

(k) μ^{11} (l) μ^{12} (m) μ^{13} (n) μ^{14} (o) μ^{15}

(p) μ^{16} (q) μ^{17} (r) μ^{18} (s) μ^{19} (t) μ^{20}

(u) μ^{21} (v) μ^{22} (w) μ^{23} (x) μ^{24} (y) μ^{25}

Figure 5.5: The mean vectors of fast walk learning sequences (hue encodes temporal offset)



(a) u_1^{01} (b) u_1^{02} (c) u_1^{03} (d) u_1^{04} (e) u_1^{05}

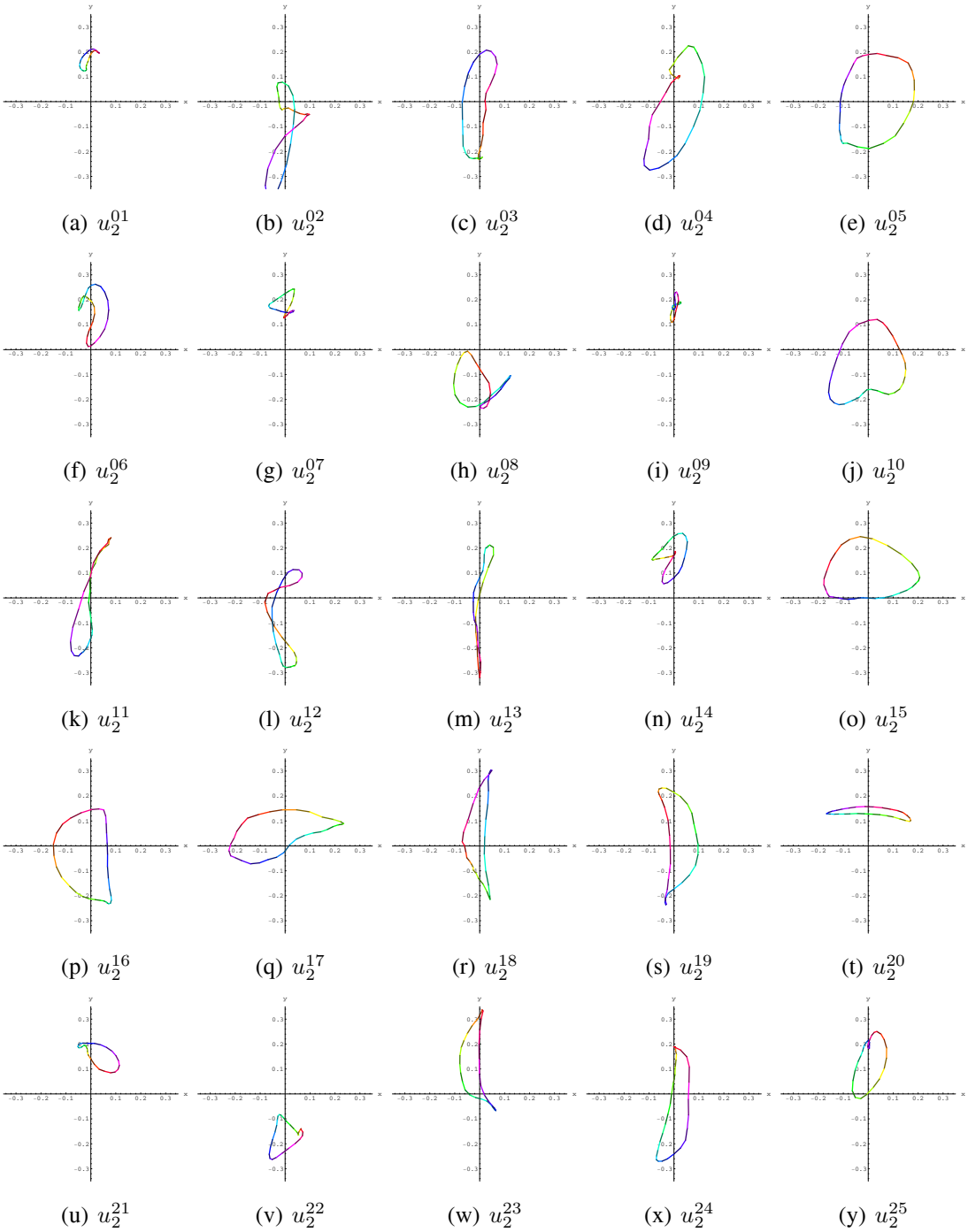
(f) u_1^{06} (g) u_1^{07} (h) u_1^{08} (i) u_1^{09} (j) u_1^{10}

(k) u_1^{11} (l) u_1^{12} (m) u_1^{13} (n) u_1^{14} (o) u_1^{15}

(p) u_1^{16} (q) u_1^{17} (r) u_1^{18} (s) u_1^{19} (t) u_1^{20}

(u) u_1^{21} (v) u_1^{22} (w) u_1^{23} (x) u_1^{24} (y) u_1^{25}

Figure 5.6: The 1st principal vectors of fast walk learning sequences (hue encodes temporal offset)



(a) u_2^{01} (b) u_2^{02} (c) u_2^{03} (d) u_2^{04} (e) u_2^{05}

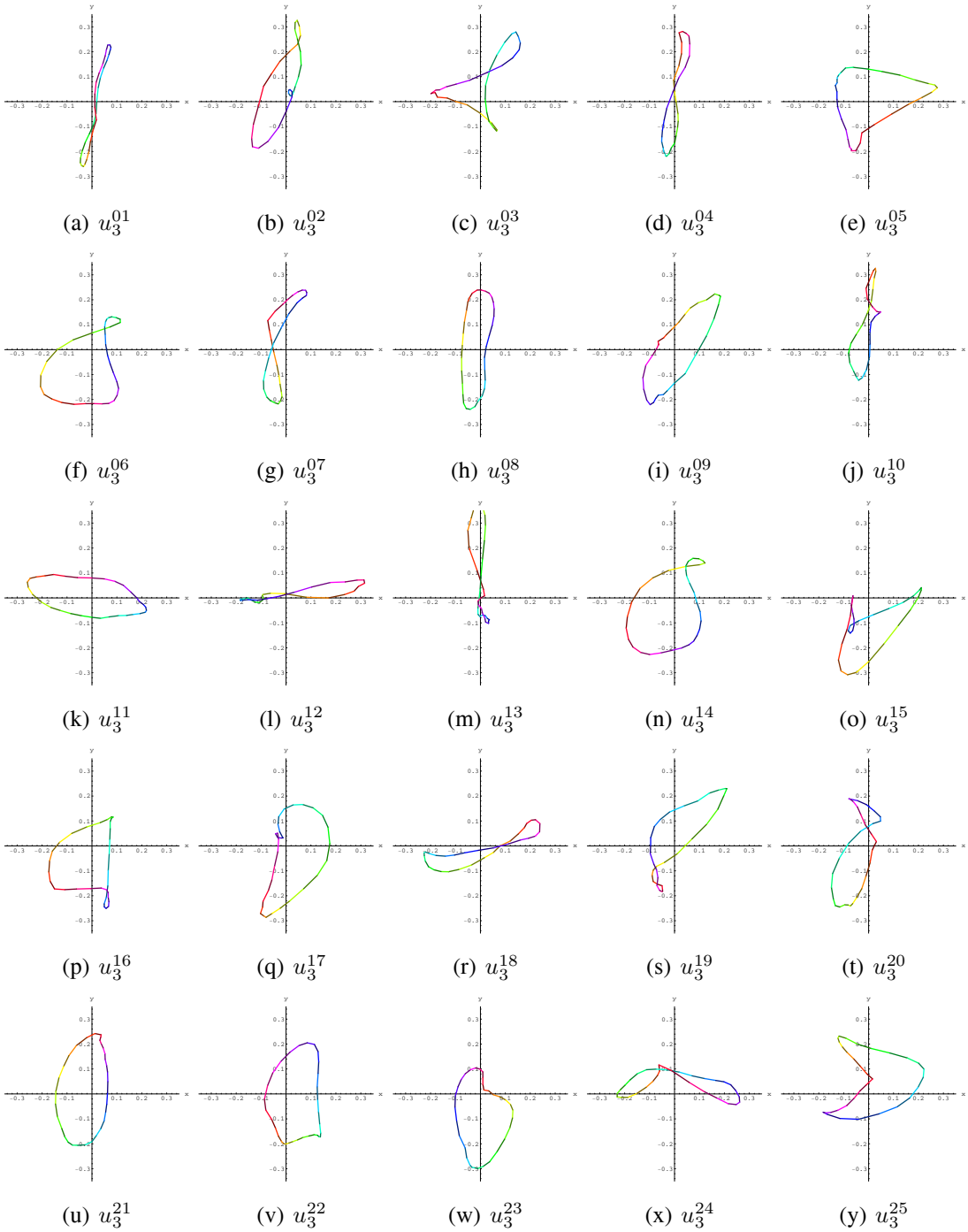
(f) u_2^{06} (g) u_2^{07} (h) u_2^{08} (i) u_2^{09} (j) u_2^{10}

(k) u_2^{11} (l) u_2^{12} (m) u_2^{13} (n) u_2^{14} (o) u_2^{15}

(p) u_2^{16} (q) u_2^{17} (r) u_2^{18} (s) u_2^{19} (t) u_2^{20}

(u) u_2^{21} (v) u_2^{22} (w) u_2^{23} (x) u_2^{24} (y) u_2^{25}

Figure 5.7: The 2nd principal vectors of fast walk learning sequences (hue encodes temporal offset)



(a) u_3^{01} (b) u_3^{02} (c) u_3^{03} (d) u_3^{04} (e) u_3^{05}

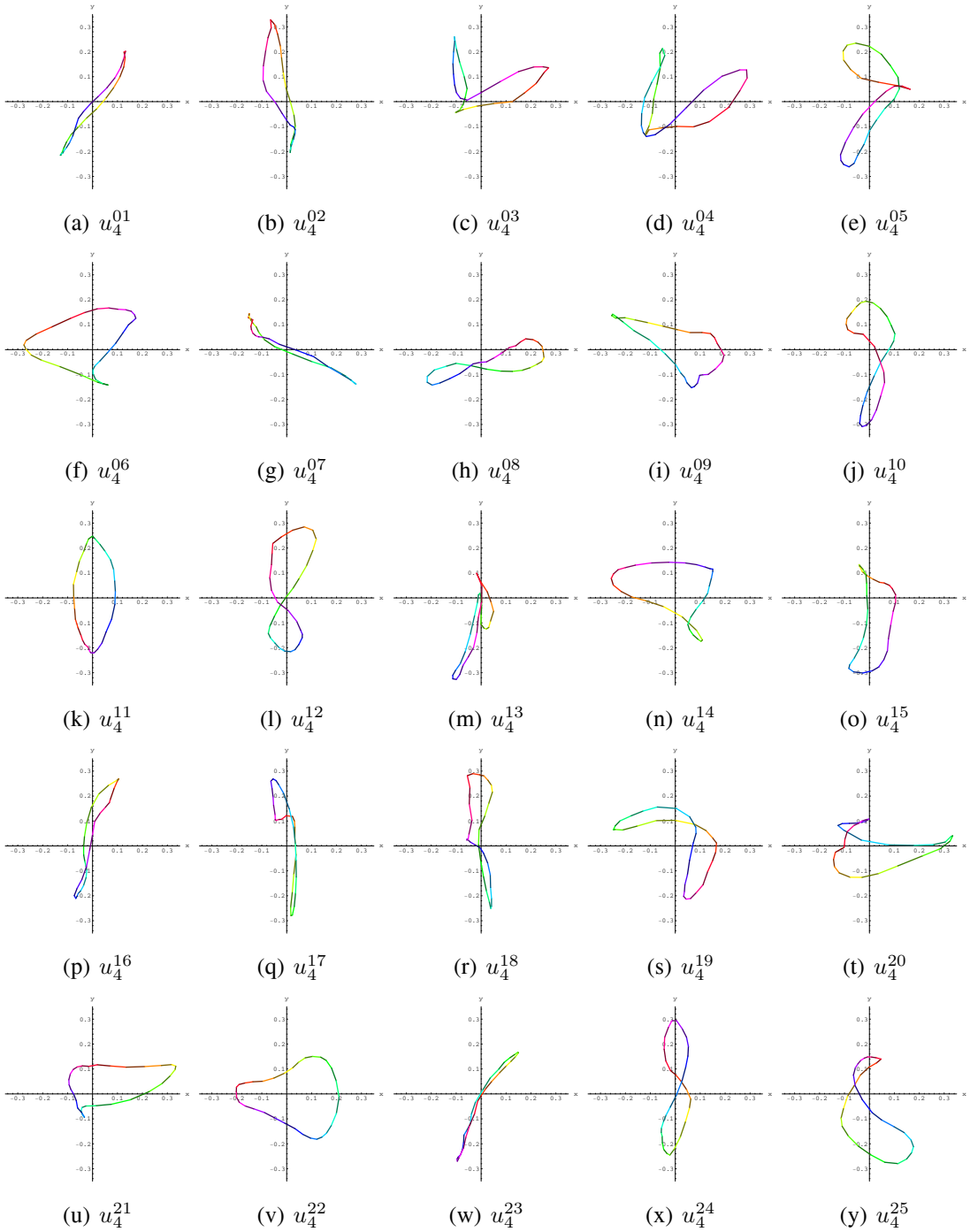
(f) u_3^{06} (g) u_3^{07} (h) u_3^{08} (i) u_3^{09} (j) u_3^{10}

(k) u_3^{11} (l) u_3^{12} (m) u_3^{13} (n) u_3^{14} (o) u_3^{15}

(p) u_3^{16} (q) u_3^{17} (r) u_3^{18} (s) u_3^{19} (t) u_3^{20}

(u) u_3^{21} (v) u_3^{22} (w) u_3^{23} (x) u_3^{24} (y) u_3^{25}

Figure 5.8: The 3rd principal vectors of fast walk learning sequences (hue encodes temporal offset)



(a) u_4^{01} (b) u_4^{02} (c) u_4^{03} (d) u_4^{04} (e) u_4^{05}

(f) u_4^{06} (g) u_4^{07} (h) u_4^{08} (i) u_4^{09} (j) u_4^{10}

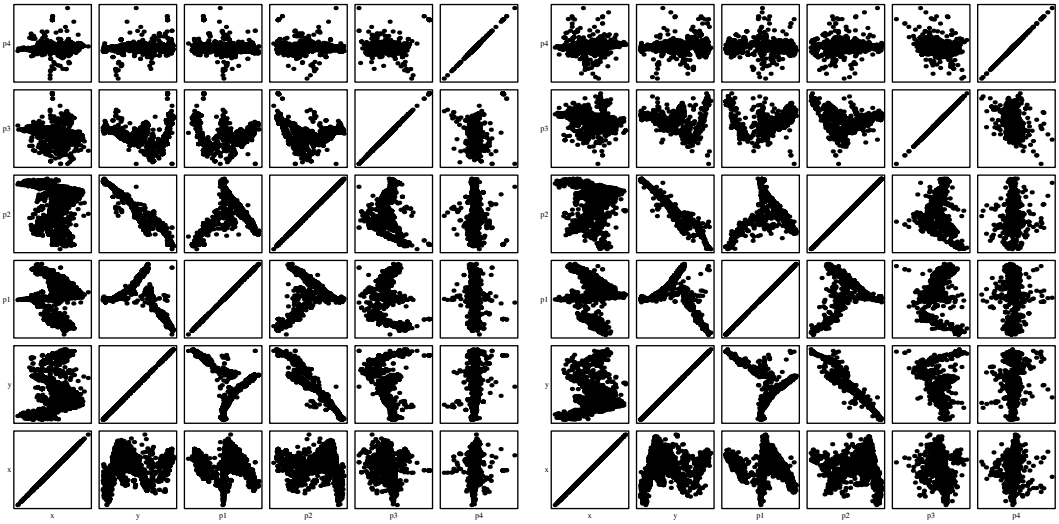
(k) u_4^{11} (l) u_4^{12} (m) u_4^{13} (n) u_4^{14} (o) u_4^{15}

(p) u_4^{16} (q) u_4^{17} (r) u_4^{18} (s) u_4^{19} (t) u_4^{20}

(u) u_4^{21} (v) u_4^{22} (w) u_4^{23} (x) u_4^{24} (y) u_4^{25}

Figure 5.9: The 4th principal vectors of fast walk learning sequences (hue encodes temporal offset)

5.2.3 Analysis of subspace distribution similarity



(a) Prior distribution A

(b) Observed distribution A

(c) Observed distribution B

(d) Observed distribution C

Figure 5.10: Scatter plots of prior and observed distributions of fast walk sequences

Figure 5.10 presents an example scatter plot of a learned prior subspace distribution (subfigure 5.10(a)), an observed distribution of the same subject (subfigure 5.10(b)) and observed distributions of two different subjects (subfigures 5.10(c) and 5.10(d)). All sets were remapped to the prior subspace for comparison.

Each box of a scatter plot shows the distribution in a two-dimensional subspace. There are 36 boxes, because the subspace used is six-dimensional. The coordinates in each box

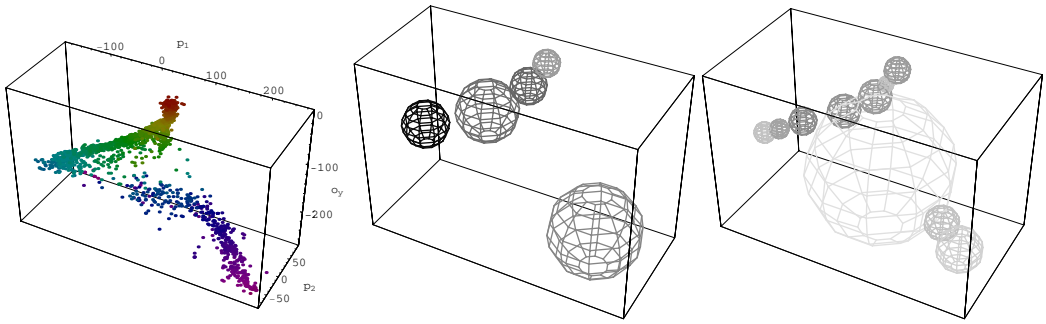
are normalized to represent the whole distribution in the respective sub-space.

The distributions of the prior (subfigure 5.10(a)) and observed distributions (subfigure 5.10(b)) of the same subject show no big differences. Subfigure 5.10(c) shows small differences in some subspaces and big differences in other subspaces. Subfigure 5.10(d) shows large differences in nearly all subspaces. There is also an obvious outlier noticeable as a solitary cluster in the first column (and the bottom row which is its mirror image).

We already represented the mappings of the distributions in 3D subspaces in Figure 3.10.

5.2.4 Choosing the number of Gaussians

First, we did some tests by animating the trajectories represented the means of Gaussians and empirically chose the number that resulted in smooth and subjectively accurate animations.



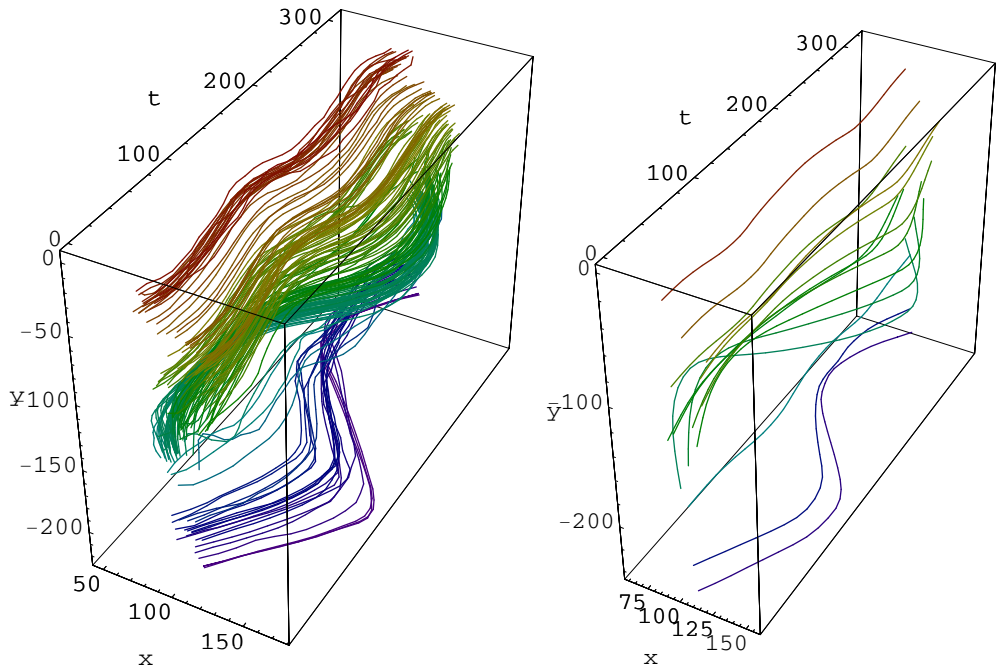
(a) ST-curve set distribution (b) GMM $K = 5$ (c) GMM $K = 10$

(d) GMM $K = 15$ (e) GMM $K = 20$

Figure 5.11: Illustration of Gaussian mixture models with varying number of Gaussians (sphere size is proportional to variance σ_k and sphere darkness is proportional to probability w_k)

We performed most experiments with $K = 15$ Gaussians, after some initial exper-

iments with motion animation suggested the subjective results were better than with $K = 10$ Gaussians, and the reconstructed animations can subjectively be recognized as very accurate generalization of cyclic locomotion. In essence, the result is analogous to Johansson light displays of cyclic locomotion, but in our case is generated automatically from the video sequence without any markers. We provide an illustration of reconstructed trajectories of 15 Gaussian means in Figure 5.12.



(a) Original set of ST-curves (every 10th curve) (b) ST-curves reconstructed from 15 Gaussian means

Figure 5.12: Spatio-temporal curves of 15 Gaussian means (hue encodes o_y)

To check the influence of the number of Gaussians we performed a test of identification from sequences of fast walk locomotion with varying number of Gaussians (the exact method is described in section 5.3.4). We illustrate the Gaussian mixture models for one of the subjects in Figure 5.11. We summarize the results in Table 5.2.

The results show our method is not very sensitive to the number of Gaussians. The most interesting result is that the recognition results degrade only slightly with 5 Gaussians. Based on our understanding of the method we interpret this result as an indication that the shape of the ST-curves is actually more important than the accurate distribution of ST-curves in the subspace. This is not too surprising, given that the shape of distribution is mostly determined by linear interpolation between the trajectories of a few joints,

K	Recognition fast walk
5	23/25
10	24/25
15	24/25
20	24/25

Table 5.2: Recognition results for varying number of Gaussians

and the distribution of ST-curves along the limb will likely cross the local mode of the distribution. However, the results could be affected by some additional distinctions in distribution density introduced by clothes and texture, or by the curve extraction method. Also, this interpretation assumes the exact scale is known!

5.2.5 Expectation-Maximization convergence

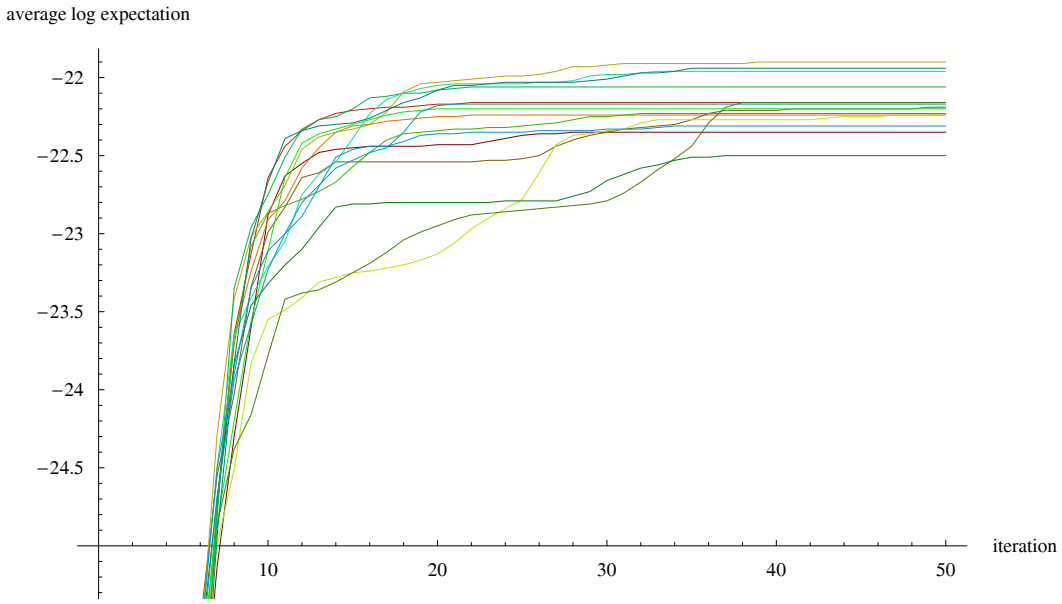


Figure 5.13: Convergence of expectation-maximization after 15 random initializations

We illustrate the convergence of Expectation-Maximization algorithm for our datasets in Figure . The results show that around 30–40 iterations are required to maximize the

expectation of our model after an initialization produced by a random subset from our dataset. It is also obvious that many runs converge to a local mode, and multiple runs need to be performed to maximize probability of arriving at a good result.

5.3 Recognition

The following sections describe experiments related to recognition of people and activities (locomotion modes) using the proposed spatio-temporal model.

5.3.1 Principal component based phase alignment

Given two sets of cyclic curves we need to phase align them to facilitate further matching. In section 3.6.1 we asserted that phase alignment can be attained by correlating principal vectors and searching for modes of correlation. Of course, there are some immediate concerns we need to answer:

- How do the principal vectors correlate practically?
- Does the maximum of *a posteriori* likelihood really correspond to one of the modes of the correlation curve?
- Is the global maximum of *a posteriori* likelihood really the one we are looking for?

Table 5.3 presents correlation results of learned and observed 1st principal vectors of the fast walk sequences. The most and least similar pairs of vectors are illustrated in Figure 5.14. The two outstanding pairs (according to the correlation criteria) have different causes. The first person was obese and he was wearing a black shirt. That caused a lot of noise which affected principal vectors (see Figure 5.14(c)). The second pair (see Figure 5.14(d)) had lower correlation due to the offset of the curve being cancelled out before correlation, the shape itself is visibly similar. The second most similar pair (see Figure 5.14(b)) exhibits different orientation of principal vectors.

We expect the first principal vectors to closely represent the longitudinal oscillation along the direction of locomotion. Thus we expect the correlation curve to loosely represent sinusoidal function with easily discernible modes. We intuitively expect maximum likelihood close to one of the maxima (or minima due to unknown vector orientation) of correlation, which would correspond to aligning the phase of oscillation. Naively looking at the principal vectors of our exemplars reveals no surprises, however practical tests demonstrate an overlooked property of likelihood.

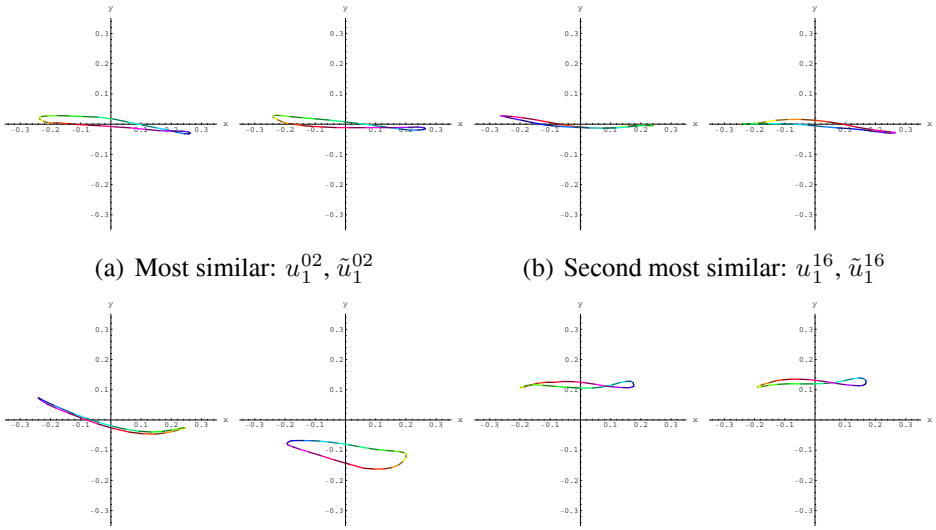
ID	correlation
1	0.79087
2	0.99848
3	0.99654
4	0.99735
5	0.99709
6	0.96292
7	0.99050
8	0.97946
9	0.99156
10	-0.96863
11	0.99228
12	0.99631
13	0.99113
14	0.97023
15	0.99582
16	-0.99760
17	0.98743
18	0.99628
19	-0.99644
20	0.54381
21	0.94198
22	-0.92837
23	0.96803
24	0.99088
25	0.99515

Table 5.3: Correlation of learned and observed 1st principal vectors of fast walk sequences

Figure 5.15 illustrates the effect of phase alignment on maximum *a posteriori* likelihood. We can notice the following:

- The correlation of principal vectors is indeed very close to a sinusoidal function. We expect this property could serve for non-PCA based phase alignment, but we did not test any such implementation, because an initial requirement was that we arrive at the appropriate basis functions by PCA.

- A local maximum of *a posteriori* likelihood does correspond exactly to either the



(a) Most similar: u_1^{02}, $\tilde{u}_1^{02}$ (b) Second most similar: u_1^{16}, $\tilde{u}_1^{16}$

(c) Second least similar: u_1^{01}, $\tilde{u}_1^{01}$ (d) Least similar: u_1^{20}, $\tilde{u}_1^{20}$

Figure 5.14: Pairs of learned and observed 1st principal vectors of fast walk sequences (hue encodes temporal offset)

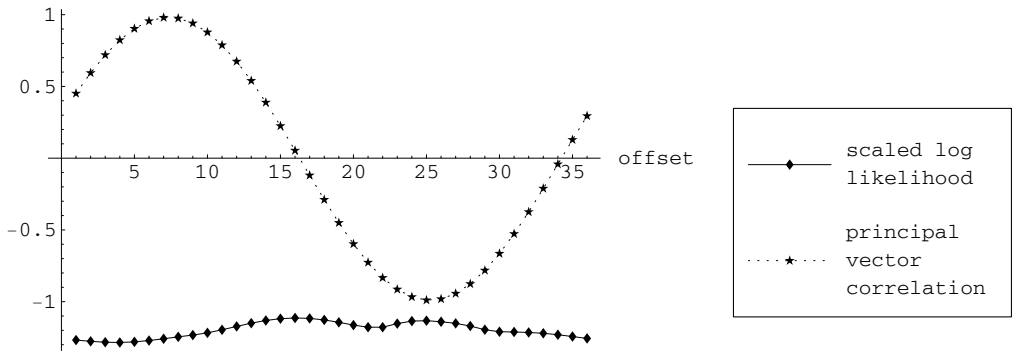


Figure 5.15: Effect of phase alignment on maximum *a posteriori* likelihood

minimum or maximum of correlation. There are two possibilities because eigendecomposition can assign either orientation to the principal vector.

- The global maximum of *a posteriori* likelihood in some cases (as in Figure 5.15) corresponds to one of the absolute minima of correlation.

So which maximum are we searching for? It turns out that the global maximum is often the wrong one. If the phase is aligned to an absolute minimum of the correlation, the curve shape vectors will more likely map close to zero in the prior subspace. These remappings

are often very close to central clusters with static position and small amplitude, which may result in increased posterior likelihood estimate and a local likelihood maximum. This happens because maximum *a posteriori* estimate on Gaussian mixture model does not penalize uncovered prior space. However, this is not the maximum we are looking for. The interesting maximum is indeed in one of the intuitively expected modes, where we have reasons to believe the maximum *a posteriori* likelihood is estimated with the correct phase alignment.

We checked the performance of PCA based phase alignment procedure experimentally. We first estimated the ground truth for the reference phase alignment. The initial estimate was the offset of the frame closest to the center of the sequence, where the person extended the left leg farthest forward. The initial estimate was done by hand. We then computed the difference matrix of phase alignment between learning and test ST-curve sets, and subtracted the phase difference of the reference frames. Even with the initial estimate, no two sets differed more than 3 units (out of 32) in phase. We then computed the average phase difference of each set against all others and updated the reference frames accordingly.

The resulting phase difference matrix is presented in Table 5.4. Ideally the matrix would be antisymmetric, but this is not possible due to quantization of phase to 32 units of the time series. The difference matrix confirms excellent performance of the proposed phase alignment procedure. Only a single test set differed from the corresponding learning set by a single unit, whereas no test set stood out by differing more than a single unit from any other learning set. This confirms that it is possible to phase align all the sets and the phase alignment procedure showed no deviation exceeding the quantization error.

5.3.2 Scale invariance

We tested for scale invariance by linearly adapting data scale S in range $[0.8, 1.2]$ and maximizing *a posteriori* likelihood. The resulting classification gave unfavorable results.

Shrinking the observation space causes the estimated likelihood to increase in most cases, because far out data points move closer to central Gaussian means. Therefore even our theoretical requirement of self-classification maximizing likelihood at a scale of 1 is not met.

We suspect an improvement of the classification method which accounts for the unmatched probability space would be required for scale invariance.

We conclude that our approach requires some other method of finding scale to achieve scale invariance.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	**0**	0	0	0	0	0	1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	**0**	0	0	0	0	1	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	1
3	0	0	**0**	-1	-1	0	0	0	0	-1	-1	-1	-1	0	-1	0	0	0	-1	-1	0	0	0	-1	1
4	0	0	1	**0**	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0	-1	1	0	**0**	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	0	0	0	-1	-1	**0**	0	0	0	-1	-1	-1	-1	0	-1	0	0	0	-1	-1	0	0	0	-1	1
7	-1	-1	1	-1	-1	0	**0**	0	-1	0	-1	0	-1	0	0	0	0	-1	0	-1	0	0	0	0	0
8	0	-1	0	0	0	0	1	**0**	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	**0**	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	0	1
10	-1	0	1	-1	0	0	0	0	0	**0**	0	0	-1	0	0	1	0	0	0	0	0	1	1	0	0
11	0	0	1	0	0	1	1	0	0	0	**0**	0	0	0	0	1	0	0	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	1	1	0	0	**0**	0	0	0	1	0	1	0	0	0	1	1	0	1
13	0	-1	1	0	0	1	1	0	0	0	0	0	**0**	0	0	1	0	0	0	0	0	0	0	0	0
14	-1	0	0	-1	-1	0	0	0	-1	0	-1	-1	-1	**0**	0	0	0	-1	-1	-1	0	0	0	-1	0
15	0	1	1	0	0	1	1	0	0	0	0	0	0	0	**0**	1	0	1	0	0	0	1	1	0	0
16	0	0	0	-1	-1	0	0	0	0	-1	-1	-1	-1	-1	-1	**0**	0	0	-1	-1	-1	0	0	-1	1
17	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	**0**	1	0	0	0	1	1	0	0
18	0	-1	0	0	0	0	0	0	0	-1	0	-1	0	0	-1	0	-1	**0**	0	0	-1	0	0	0	0
19	0	-1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	**0**	0	0	0	0	0	0
20	-1	-1	1	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	**0**	0	0	1	0	0
21	-1	0	0	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	0	-1	-1	**-1**	0	0	-1	-1
22	-1	-1	0	-1	0	-1	0	0	0	-1	-1	-1	-1	0	-1	0	-1	-1	0	0	-1	**0**	0	0	0
23	0	0	0	0	0	0	1	0	0	-1	0	-1	0	-1	-1	0	-1	0	0	-1	-1	0	**0**	0	0
24	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	**0**	1
25	-1	-1	0	-1	0	-1	0	0	0	0	-1	-1	-1	0	0	0	0	-1	0	0	0	0	0	0	**0**

Table 5.4: Phase difference matrix of learned and observed 1st principal vectors of fast walk sequences

5.3.3 Approximate spatial alignment

Based on the observation that the ST-curve distribution data in a subspace exerts most variance in the o_y coordinate and first principal component p_1, we experimented with an implementation of approximate spatial alignment by a linear transform based on these two components only.

We initialize the remapping two times as identities with alternative signs for p_1. Initial estimate is improved by a simulated annealing procedure, the metric used is the sum of squared distances to nearest neighbor computed for each point of either model against the other model. An approximation of linear remapping of the distributions is attained by minimizing the metric.

In the examples, we noticed that the approximate matching at this point succeeds for locomotion of different subjects captured from the same angle, but if we relax linear transform to stretch principal components independently, we can also approximately match distribution of locomotion captured from slightly different angles. We tested for a 45° angle, however the procedure would probably succeed for all angles where the principal axis contains the majority of energy in the direction of motion. One of the experimental results of remapping between two different persons is depicted in Figure 5.16.

It might be possible to find spatio-temporal correspondences among superpositions of Gaussians by developing a more advanced algorithm from this idea.

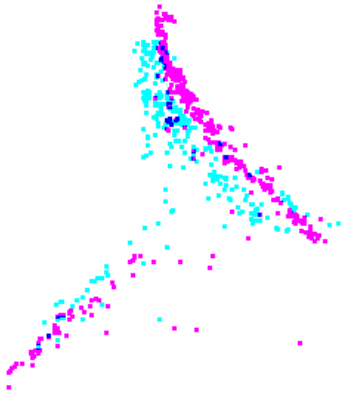


Figure 5.16: Approximate remapping of ST-curve distributions in 2-dimensional subspace

mode	view	sequences	recognition errors	order of correct class
fast walk	side-view	25	1	2nd
slow walk	side-view	25	0	
incline walk	side-view	25	1	2nd
ball walk	side-view	24	0	
slow walk	back-view $45°$	25	1	3rd

Table 5.5: Summary of recognition results

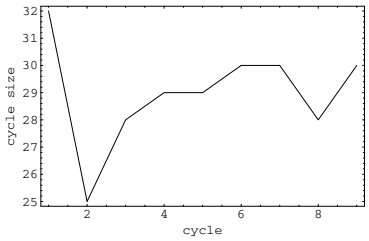
5.3.4 Identification

We performed PCA and kept 4 principal components to represent spatio-temporal variation. The data vectors thus contained 2 spatial parameters representing ST-curve centroid in the viewspace and 4 spatio-temporal parameters representing ST-curve shape. We trained a diagonal GMM with 15 Gaussians using EM algorithm initialized on a random subset of data vectors with random variance scaled from data interval. We used 30 EM iterations on 10 random initializations and kept the best GMM that maximized expectation for each set of the training vectors.

We tested the classification of 25 test sequences against 25 training sequences using MAP estimate. We phase aligned the sequences by maximizing correlation of first principal vectors of both sequences. We remapped observation vectors to learned prior space using only 4 principal components. We tested for different spatial offsets by performing exhaustive search in range $[-32...32]$ for x and $[-16...16]$ for y axis with a step size of 4.

The identification results are summarized in table 5.5. Our method correctly classified

(a) A gesture (b) Positional variations



(c) Cycle length variation

Figure 5.17: Probable causes of recognition errors

people in nearly all cases. We inspected the misclassified sequences. We noticed that one misclassified sequence contained deviant arm gestures in the training part of the sequence. The other misclassified sequence contained non-cyclic arm movement and very similar leg movement.

For each mode we illustrate the classification results as a graphical diagram illustrating maximum *a posteriori* likelihood of prior model classes given the ground truth class of the observed data. Thus each column in the diagrams shows the relative probability of all the classes for a given observed class. The class in the top is the chosen classification result. Average log likelihood of the dataset is displayed instead of the computed log likelihood to normalize the effect of different number of curves in the observed dataset for an easier comparison of results between classes.

The figures show results for side views of fast walk (Figure 5.18), slow walk (Figure 5.19), incline walk (Figure 5.20), ball walk (Figure 5.21), and back-side view of slow walk (Figure 5.22).

Figure 5.18: Recognition results: fast walk

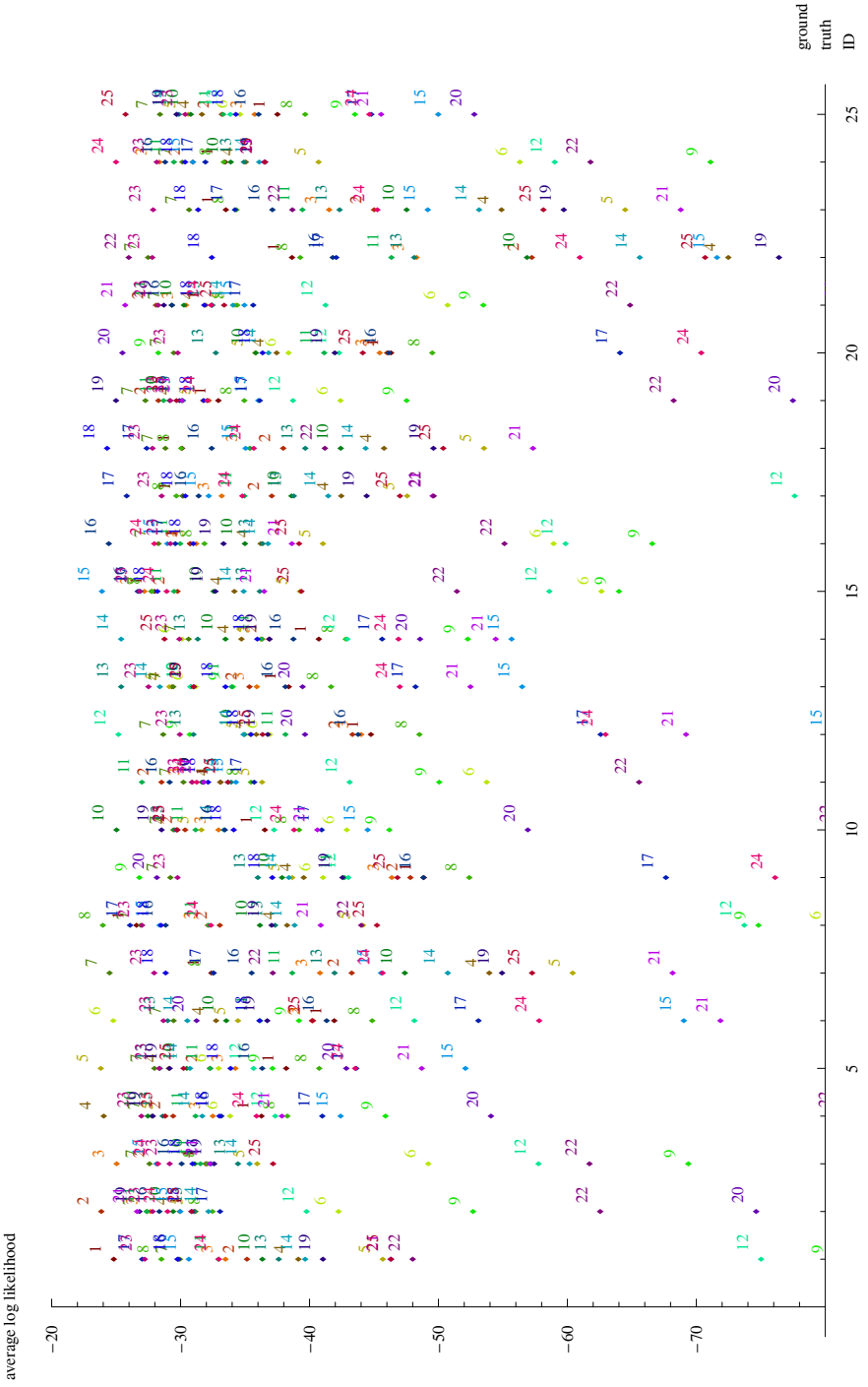


Figure 5.19: Recognition results: slow walk

Figure 5.20: Recognition results: incline walk

Figure 5.21: Recognition results: ball walk

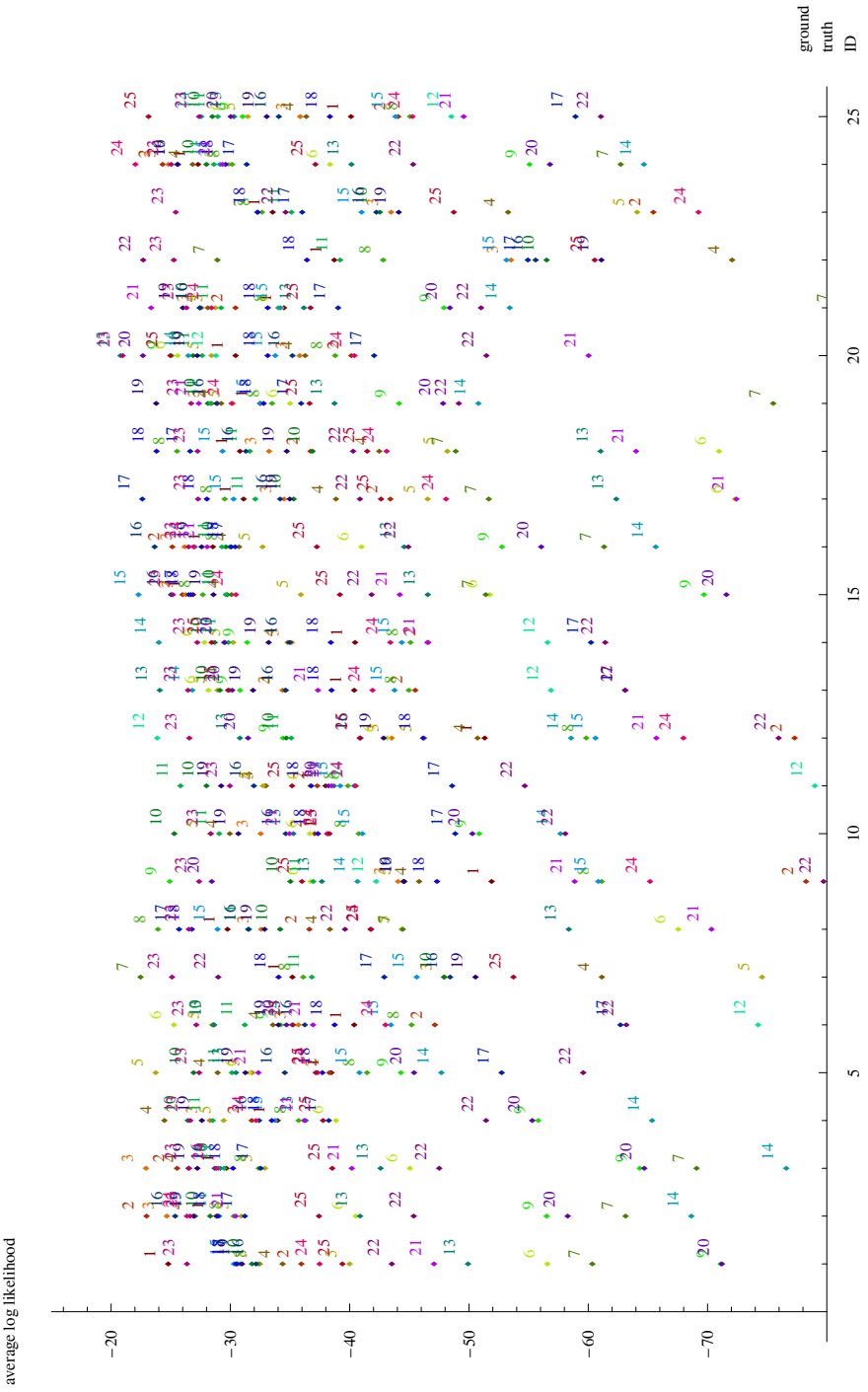


Figure 5.22: Recognition results: slow walk - 45° back-view

5.3.5 Identity and Activity recognition

The purpose of this experiment is to demonstrate the applicability of our method to simultaneous identity and activity recognition, as well as to illustrate class separation. The class label is composed of person ID and locomotion mode. We test a compositum of 25 persons in 4 locomotion modes (fast walk - F, slow walk - S, incline walk - I, and ball walk - B), except for person 25 in ball walk mode, which was not present in our database.

The MAP classification results are presented in Figure 5.23. To summarize, our method successfully classified all but 2 instances of locomotion.

One of the erroneous examples (22f) was already illustrated in Figure 5.17. In the single-mode test it was classified to 23f, due to a gesture in the learning sequence. In this test, it was classified to 23b, suggesting similar leg motion was enough for similar re-classification (together with bad prior of 22f).

The second misclassification of 22i to 23i again involved the same persons. Besides similar leg motion we believe cycle size variation and significant locomotion variations in 23i (see Figure 5.17(b)) contributed to misclassification.

Figure 5.23: Recognition results: Identity and Activity

5.4 Testing in uncontrolled environment

The main purpose of these tests was not to evaluate the performance of our implementation, but to identify the issues to be solved in order for the methods to be used out of the laboratory, and more generally, to illustrate the spatio-temporal features for the design of further theoretical approaches.

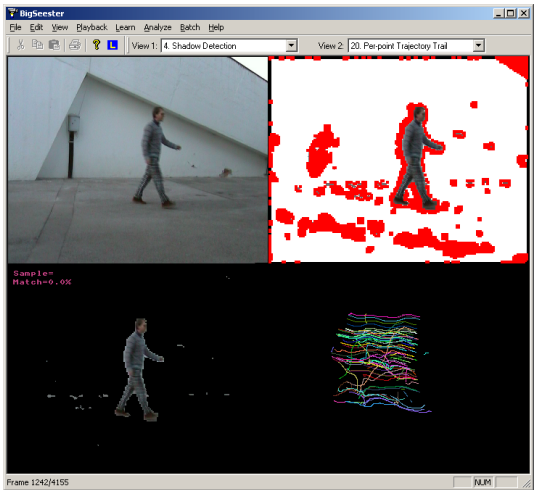
The setting for the tests was an outdoor pedestrian area in the Kranj stadium complex. We found an area with static background and a lot of pedestrian traffic. The recordings were done in the late afternoon, with some variation in illumination due to transient clouds and nearing sunset.

We found the following issues need to be addressed before our learning and recognition methods could be used in uncontrolled environment:

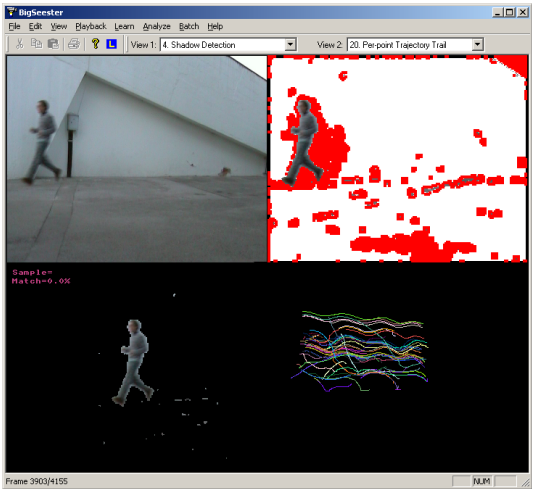
- The coordinate reference frame was static in the laboratory. We would need to track the subject and filter motion trajectories to remove translation.
- The depth and therefore size was constant in the laboratory. We would need a method to estimate the scale of the observed subject. The scale must be normalized at least for learning, but it is also helpful for recognition. Also, we use constant patch size, which would need to be adapted based on the subject size for best performance/complexity ratio.

We conclude that at least the following issues need to be addressed to generalize similar markerless spatio-temporal approaches to learning and recognizing locomotion:

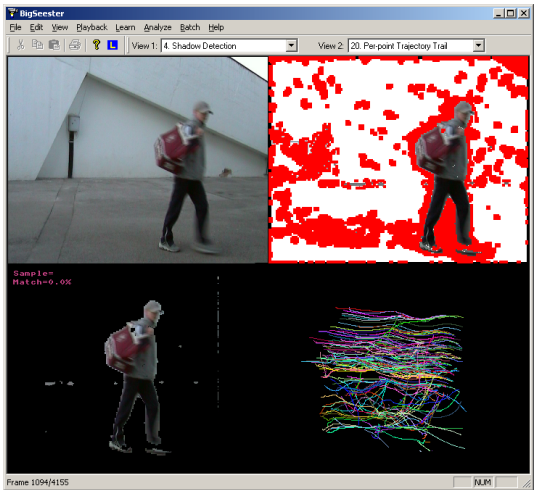
- view-dependence,
- cycle size variations and non-cyclic movements,
- mode and orientation transitions,
- combining tracking and spatio-temporal representation,
- segmentation, handling multiple subjects and interactions.



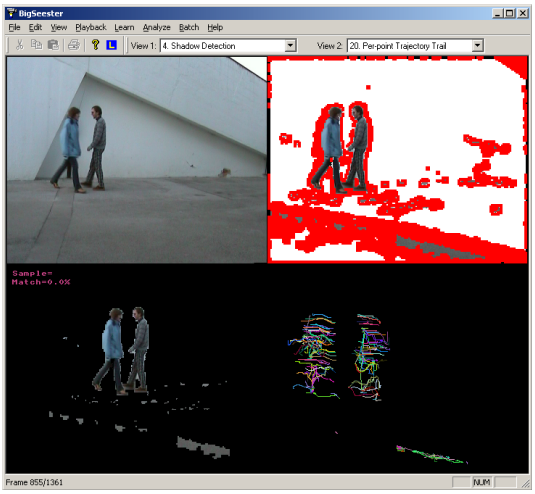
(a) image 12



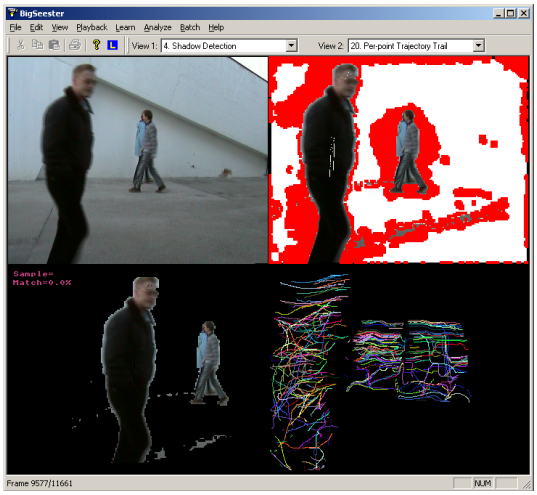
(b) image 12a



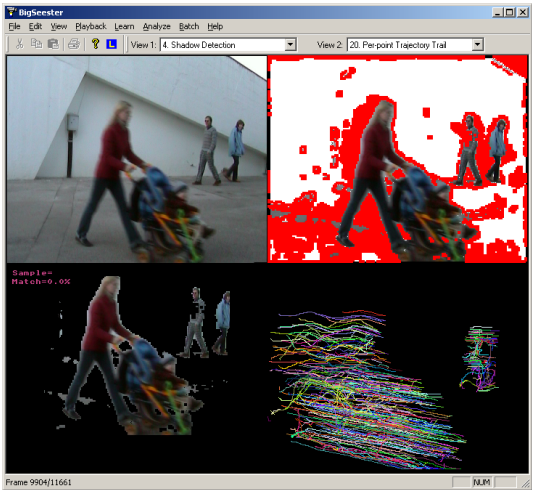
(c) image 5



(d) image 24



(e) image 21



(f) image 22

Figure 5.24: Testing in uncontrolled environment

Chapter 6

Discussion

6.1 Further possible extensions

The results confirm our basic idea that human locomotion can automatically and efficiently be modeled, learned and recognized by a Gaussian mixture model of the distribution density of spatio-temporal curves in an automatically learned optimal subspace.

However, based on our understanding of the method, we feel the need for further testing to discount the influence of scale, clothes, texture dependence and cycle variation dependence, before advocating the method as a general approach to gait recognition. We believe that errors in scale and coordinate reference could be critical for practical performance.

The method is view dependent, but it can theoretically include some view variation in the prior probabilistic model.

The main reason for scale dependence is the fact that as the scale gets smaller and the distribution tends to approach the central Gaussian means increasing estimated likelihood. The work around would be a better method of matching prior and observed distributions to account for unexplained prior Gaussians. We could also use some other means of scale detection (silhouette size, energy estimation) and normalize the datasets accordingly.

Texture influences the density of point-tracking. Low texture increases the probability of losing point tracks which decreases uniformity of distribution density over space, which can introduce artificial differences in models of locomotion. This does not matter though if we observe exactly the same object over time. An alternative approach could use an interest operator to find stable features for tracking, which would overcome some of the problems with texture dependence, but it would also introduce some new problems in cases of sparse and nonuniform sampling.

Clothes' textures are the reason for texture dependence and additional dynamics of loose clothes. Our model can include this dynamics, but it would introduce additional differences.

Cycle size variation influences curve stitching. The problem can be solved by the method of trajectory extraction. It must deal with varying cycle sizes and stitch at different offsets in cycle interval to avoid phase-dependence due to single-offset error accumulation.

Lastly, there is a time dependence: gait changes over time due to fitness, energy level, mood, injuries, shoes and many other factors. We could include all these factors by building a prior model from cycles of several sequences recorded with such variation, but then the model would be less specific and we would need to rerun recognition tests.

6.2 Knowledge transfer

We investigate opportunities for knowledge transfer in the light of results and their interpretation.

From controlled to uncontrolled environment

The transfer of models from controlled to uncontrolled environment is trivial; *i. e.* we can automatically and relatively robustly learn priors in controlled environment in an unsupervised way. The method can tolerate some noise and outliers during learning (we had noise in many learning sequences) and it can tolerate even more noise and outliers during recognition due to probabilistic formulation.

The method itself would have to include tracking of the subject, in order to arrive at the reference frame required for learning and recognition.

From person to person

The transfer of models from person to person is practical in two ways:

1. Given similarity of some activities between two persons we can assume other activities are similar and apply corresponding priors.

2. Theoretically we can learn general priors but we would need to accumulate loops from a lot of test subjects. We could cluster subjects by similarity and build quite accurate general models for a range of human sizes, postures and locomotion styles.

From activity to activity

The transfer of models from activity to activity is problematic, because we need to have some information about spatial and structural correspondences. In principle, the transfer could be implemented in a similar way to transferring data from markers to skeletal models in professional 3D graphics applications.

Gaussian clusters have a spatial correspondence. We would need to establish correspondence between clusters and body position. There has been some work on the topic for the case of single labeled markers in 3D, however there is no prior work for the case of random tracked points of locomotion in 2D view-space. We have implemented only a simple correspondence experiment as a test of concept.

It should be possible to interpolate clusters between similar activities (*e. g.* between slow walk and fast walk). It should be possible to change only some clusters to alter activity models. It is possible to recognize similar activities despite changing some clusters (locomotion with gestures etc.). In fact the latter was accidentally experimentally confirmed by recognizing fast walk with a gesture as most similar to ball walk of the same person.

6.3 On extensions and alternative implementations

We propose possible alternative implementations or extensions and outline directions for our future work.

6.3.1 Distribution matching

One of the weak points of the recognition method is the way prior and observed distributions are matched. The implemented method of recognition linearly remaps observed data to prior subspace and computes maximum *a posteriori* likelihood given a Gaussian mixture. However, the fact that some part of probability space of prior distribution is not covered by the observed data is never accounted for. It does not decrease likelihood estimate. Theoretically, we could accidentally map all datapoint close to the maximum of distribution density and we would get the highest likelihood estimate, though the majority of prior distribution density would not be observed at all.

A similar problem of lesser scale becomes evident when mapping compact distribution modes to sparse distribution modes. If a compact distribution has similar modes to a sparse distribution, it's possible that the compact model has a higher MAP likelihood

estimate than the sparse model. A compact distribution can map to some of the highly probable modes of the sparser distribution, when the rest of the prior distribution is not covered, but *a posteriori* likelihood is still higher.

Thus it is of ultimate importance to improve the distribution matching method.

Two-way distribution testing

A straightforward solution to the distribution matching problem would be two-way distribution testing. The basic idea is to build a model of observed distribution, and test both ways against the prior to determine overlap. We can implement this by either proceeding by estimating maximum *a posteriori* likelihood of observed data given prior model and *vice versa*, or by computing a product of Gaussian distributions (however in the latter case the quality of Gaussian modeling may become very important).

Deterministic methods

In alternative to a global probabilistic method of building priors, which may need a lot of iterations to arrive at a useful result, we could also explore deterministic methods of building priors.

We could represent each prior datapoint probabilistically by a radial basis function, based on some quantitative metric of the neighborhood. We could model prior distribution by deterministic clustering and assigning radial basis functions to the clusters. Thus the method of building models would be deterministic but the recognition could still be probabilistic.

6.3.2 Overcoming continuity requirement

The current method only extracts curves of continuously trackable points, however there is additional information about locomotion contained in the curves of the surface points that appear only temporally. We think that the relative motion of all the limb tips would improve the locomotion model significantly.

In order to overcome continuity requirement, we would first need and alternative tracker. We could interpolate short breaks. We could find long segments and stitch them to a cycle by interpolating at the missing section. We could distinguish appearance features and track by appearance.

We could use partially tracked curves and apply a robust PCA method [13] to deal with the missing data.

3D

The extension of our method to 3D is trivial. Instead of 2D point tracker we need to apply a 3D point tracker, and the extension of spatio-temporal model only requires extension of 2D view-space coordinate vectors to 3D coordinates. All the developed methods should work in principle.

We see an opportunity to learn priors in 3D and project the model to 2D view-space for tracking and recognition tasks. In fact we would recommend this method to efficiently learn priors in a laboratory. We could also automatically build piecewise view-specific locomotion priors for tracking from lab-made 3D priors!

6.4 Future work

Instead of perfecting this very specific method of learning and recognizing locomotion models, it is probably better to take a broader view of how the principles can be extended and make the methods useful in more practical setting, and how the developed methods can be combined with other methods to make advances in visual learning and recognition from video sequences. In the future we intend to apply spatio-temporal features to build more general visual phenomena models.

For the purposes of locomotion modeling, at least the following issues need to be addressed:

- We need a method to include multiple views in a single framework. A simplistic solution would be to embed our model in a state model.

- We need to model non-cyclic motion, which could be achieved by piecewise ST-curve set modeling and some method of state transitions.

- There are possibilities of combining spatio-temporal trajectories and appearance, by either merging subspaces or multi-modal handling of features.

- Finally, the holy grail would be a method of on-line learning and recognition, including ability to continuously and autonomously learn, organize, and recognize locomotion.

6.5 Conclusion

We proposed a novel spatio-temporal model for representation of cyclic human locomotion in monocular view space, together with methods for learning and recognition.

The results for motion based human recognition on a the CMU MoBo database are favorable. Based on these results we argue that probabilistic modeling of locomotion based on local spatio-temporal trajectories is a useful approach to learning and recognition of locomotion models for recognition and tracking.

Tests in outdoor environment illuminate numerous outstanding issues to be resolved in order to apply or generalize the method, before our visions can be realized.

The developed method is marker-less and structure-free and could be applied to learning and recognition of other types of cyclic motion. We believe we are the first to successfully learn and recognize such a model from random local spatio-temporal features only.

Our method of representing sets of spatio-temporal curves can be seen as a novel spatial generalization of single principal curves. The method is universal and could be applied to learning other sets of spatio-temporal curves.

Appendix A

Principal component analysis

Principal Components Analysis or the Karhunen-Loeve expansion is a classical method for dimensionality reduction, and a chapter on the subject may be found in numerous texts on multivariate analysis. One reference among many is Anderson [3].

PCA was first formulated by Hotelling in 1933. Examples of its many applications include data compression, image processing, visualization, exploratory data analysis, pattern recognition and time series prediction.

A.1 Basics of PCA

Principal component analysis is a linear transformation from a high-dimensional input space to a low-dimensional feature space, which among all linear transformations guarantees the best possible representation of the high-dimensional input vectors in the low-dimensional feature space. It rotates the coordinate frame in a data-driven way, such that the variability of the input data can be efficiently described using only a small number of basis vectors.

The principal axes are obtained in such a way that they minimize the squared reconstruction error between the input points and their representations and they maximize the variance of the principal components. It turns out that these two criteria, namely the minimization of the reconstruction error and the maximization of the variance, are equivalent and can be uniquely satisfied using PCA.

PCA produces very good results, if the high-dimensional input vectors are correlated. This means that they contain redundant information. PCA removes the redundancy by decorrelating the input vectors; the new coordinates of the input vectors (principal components) are uncorrelated. As a consequence, the correlated high-dimensional input vectors

can be efficiently represented as the uncorrelated low-dimensional vectors of principal components making PCA a very powerful tool for data compression.

A.2 Derivation of PCA

After the initial informal outline of PCA, we will now derive and describe PCA more formally. First we will derive PCA by maximizing the variance in the direction of principal vectors. Let us suppose that we have N M-dimensional vectors $\boldsymbol{x}_j$ aligned in the data matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$. Let $\boldsymbol{u}$ be a direction (a vector of length 1) in $\mathbb{R}^M$. The projection of the j-th vector $\boldsymbol{x}_j$ onto the vector $\boldsymbol{u}$ can be calculated in the following way:

$$\boldsymbol{a}_j = \langle \boldsymbol{x}_j, \boldsymbol{u} \rangle = \boldsymbol{u}^T \boldsymbol{x}_j = \sum_{i=1}^{M} \boldsymbol{u}_i \boldsymbol{x}_{i,j} \tag{A.1}$$

We want to find a direction $\boldsymbol{u}$ that maximizes the variance of the projections of all input vectors $\boldsymbol{x}_j$, $j = 1...N$. It follows that the mean of the projections is

$$\hat{a} = \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{a}_j = \frac{1}{N} \sum_{i=1}^{M} \boldsymbol{u}_i \boldsymbol{\mu}_i \tag{A.2}$$

and the variance is

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^{N} (\boldsymbol{a}_j - \hat{a})^2 = \boldsymbol{u}^T C \boldsymbol{u}. \tag{A.3}$$

Here, $\boldsymbol{\mu}_i$ is the mean of i-th row in the data matrix $\boldsymbol{X}$ and $\hat{\boldsymbol{x}}_{ij}$ is the value of $\boldsymbol{x}_{ij}$ with subtracted $\boldsymbol{\mu}_i$. If the vector $\boldsymbol{\mu}$ contains all row means, thus

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_D]^T = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \tag{A.4}$$

then

$$\hat{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{\mu} 1_{1 \times N} \tag{A.5}$$

and $\boldsymbol{C}$ is the covariance matrix of $\boldsymbol{X}$, thus

$$\boldsymbol{C} = \frac{1}{N} \hat{\boldsymbol{X}} \hat{\boldsymbol{X}}^T \tag{A.6}$$

Our goal is to maximize σ^2 under the constraint that $\|\boldsymbol{u}\| = 1$. Therefore, by using the technique of Lagrange multipliers, we have to maximize the function

$$F(\boldsymbol{u}; \lambda) = \boldsymbol{u}^T \boldsymbol{C} \boldsymbol{u} - \lambda(\boldsymbol{u}^T \boldsymbol{u} - 1) = \sum_{i=1}^{M} \sum_{j=1}^{M} \boldsymbol{u}_i \boldsymbol{c}_{ij} \boldsymbol{u}_j - \lambda(\sum_{i=1}^{M} \boldsymbol{u}_i^2 - 1). \tag{A.7}$$

A closed form solution of this maximization problem can be obtained in the following way:

$$\begin{aligned}\frac{\partial F}{\partial \boldsymbol{u}_i} = \sum_{j=1}^{M} \boldsymbol{c}_{lj}\boldsymbol{u}_j + \sum_{j=1}^{M} \boldsymbol{u}_i\boldsymbol{c}_{il} - 2\lambda\boldsymbol{u}_i = &\quad 0 \quad ; l = 1...M \\ \sum_{i=1}^{M} \boldsymbol{c}_{li}\boldsymbol{u}_i = &\quad \lambda\boldsymbol{u}_l \quad ; l = 1...M \\ \boldsymbol{C}\boldsymbol{u} = &\quad \lambda\boldsymbol{u} \quad . \end{aligned} \tag{A.8}$$

Therefore, to find $\boldsymbol{u}$ and λ that maximize (A.7) we have to compute the eigenvectors and the eigenvalues of the covariance matrix $\boldsymbol{C}$. The largest eigenvalue equals the maximal variance, while the corresponding eigenvector determines the direction with the maximal variance. By performing eigenvalue decomposition (EVD) or singular value decomposition (SVD) of the covariance matrix $\boldsymbol{C}$ we can diagonalize $\boldsymbol{C}$

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{A.9}$$

in such a way that the orthonormal matrix $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_N] \in \mathbb{R}^{M\times N}$ contains the eigenvectors $\boldsymbol{u}_1, ..., \boldsymbol{u}_N$ in its columns and the diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{M\times N}$ contains the eigenvalues $\lambda_1, ..., \lambda_N$ on its diagonal. We will assume that the eigenvalues and the corresponding eigenvectors are arranged with respect to the descending order of the eigenvalues, thus $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$. Therefore, the most of the variability of the input random vectors is contained in the first eigenvectors. Hence, the eigenvectors are called principal vectors (also principal axes or principal directions).

A.3 Properties of PCA

The orthonormal matrix $\boldsymbol{U}$ containing the principal vectors can serve as a linear transformation matrix for projection from the high-dimensional input space to the low-dimensional feature space and vice versa. The columns of $\boldsymbol{U}$ are the basis vectors of the new low-dimensional coordinate frame expressed with the high-dimensional coordinates. Thus an input vector can be projected into the principal subspace using the transformation matrix $\boldsymbol{U}^T : \mathbb{R}^M \rightarrow \mathbb{R}^N$:

$$\boldsymbol{a} = \boldsymbol{U}^T\hat{\boldsymbol{X}}. \tag{A.10}$$

Thus, the coefficients $\boldsymbol{a}_j$ are computed as the projections of the input image onto each principal vector:

$$\boldsymbol{a}_j = \langle\hat{\boldsymbol{x}}, \boldsymbol{u}_j\rangle = \sum_{i=1}^{M} \boldsymbol{u}_{ij}\hat{\boldsymbol{x}}_i \; ; j = 1...N. \tag{A.11}$$

All the input vectors contained in the input matrix $\hat{\boldsymbol{X}}$ can thus be projected as $\boldsymbol{A} = \boldsymbol{U}^T\hat{\boldsymbol{X}}$. It can be shown that since $\boldsymbol{A}$ is an orthonormal transformation of the mean centered $\hat{\boldsymbol{X}}$, the principal components are also centered around zero:

$$\boldsymbol{\mu}_A = 0 \tag{A.12}$$

and the correlation matrix of $\boldsymbol{A}$:

$$\boldsymbol{C}_A = \boldsymbol{\Lambda} \tag{A.13}$$

Therefore, the covariance matrix of the transformed data is the diagonal matrix $\boldsymbol{\Lambda}$, which contains the eigenvalues on its diagonal. This fact has two important implications. First, it proves that the transformed vectors are uncorrelated. Thus the redundancy caused by the correlation between the input vectors has been removed. Secondly, it shows that the variance in the direction of the i-th principal axis (the variance of the i-th principal components) is equal to the i-th eigenvalue λ_i, thus $\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{a}_{ij}^2 = \lambda_i$.

An important property of the diagonalization (A.9) is that it preserves the trace of the matrix which is being diagonalized. Since the sum of the diagonal elements of the covariance matrix is the sum of variances of the input vectors, this implies that the total variance of the input data has been preserved and equals the sum of all eigenvalues:

$$\text{VAR}(\boldsymbol{X}) = \sum_{i=1}^{N}\lambda_i = \text{VAR}(\boldsymbol{A}) \tag{A.14}$$

Now we will explain how $\boldsymbol{U}$ can serve as a transformation matrix for projection of the coefficient vector back into the input space. This operation is called *reconstruction*. The coefficient vector $\boldsymbol{a}$ is reconstructed using the transformation matrix $\boldsymbol{U} : \mathbb{R}^N \rightarrow \mathbb{R}^M$:

$$\hat{\boldsymbol{y}} = \boldsymbol{U}\boldsymbol{a} = \sum_{j=1}^{N}\boldsymbol{a}_j\boldsymbol{u}_j. \tag{A.15}$$

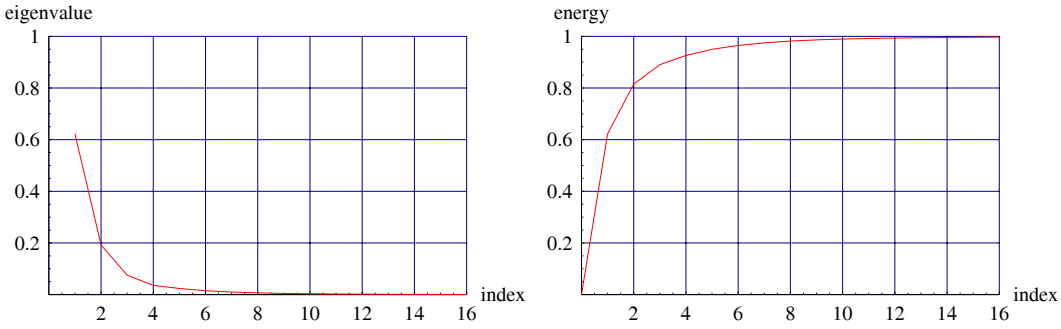
Since N eigenvectors composing $\boldsymbol{U} \in \mathbb{R}^{M\times N}$ span the same subspace in $\mathbb{R}^M$ as all N input vectors composing $\boldsymbol{X} \in \mathbb{R}^{M\times N}$, each input vector from $\boldsymbol{X}$ can be perfectly reconstructed without any reconstruction error. What is more interesting to us, is how well an input vector is reconstructed from a subset of principal components only.

To realize this, we first consider how the variance is distributed across the principal axes. This distribution is called the *eigenspectrum* and it is practically a plot of eigenvalues sorted in the decreasing order. A typical eigenspectrum is depicted in Figure A.1(a). As one can observe, most of the variance is contained in the first few eigenvectors. This can also be measured with *energy*, which is defined as a fraction of the total variance. The

energy contained in the first k eigenvectors can thus be calculated as

$$en_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{A.16}$$

The energy plot obtained from the eigenvalues depicted in Figure A.1(a) is shown in Figure A.1(b). Again, it is evident that most of the energy is contained in the first few eigenvectors already.



(a) Eigenspectrum (b) Energy

Figure A.1: Typical eigenspectrum and energy of a set of spatio-temporal curves

From this we can conclude that we can obtain a good approximation of the input images by considering only a subset of eigenvectors associated with the largest eigenvalues. Therefore, from now on, we will consider only k, $k \ll N$, principal axes, thus $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_k] \in \mathbb{R}^{M \times k}$. Now, an input vector is projected into the k-dimensional principal subspace using the transformation matrix $\boldsymbol{U}^T : \mathbb{R}^M \rightarrow \mathbb{R}^k$:

$$\boldsymbol{a} = \boldsymbol{U}^T \hat{\boldsymbol{x}} = \boldsymbol{U}^T (\boldsymbol{x} - \boldsymbol{\mu}) \tag{A.17}$$

$$\boldsymbol{a}_j = \langle \hat{\boldsymbol{x}}, \boldsymbol{u}_j \rangle = \sum_{i=1}^{M} \boldsymbol{u}_{ij} \hat{\boldsymbol{x}}_i = \sum_{i=1}^{M} \boldsymbol{u}_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_i) \; ; j = 1...k \tag{A.18}$$

and reconstructed using the transformation matrix $\boldsymbol{U}^T : \mathbb{R}^k \rightarrow \mathbb{R}^M$:

$$\hat{y} = \boldsymbol{U}\boldsymbol{a} = \sum_{j=1}^{k} \boldsymbol{a}_j \boldsymbol{u}_j \tag{A.19}$$

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{\mu}. \tag{A.20}$$

Thus, an input vector is approximated with a linear combination of the first k principal vectors. The reconstruction error (residual vector) is equal to the difference between the input and the reconstructed vector:

$$\boldsymbol{e} = \hat{\boldsymbol{x}} - \hat{\boldsymbol{y}} = \sum_{j=1}^{N} \boldsymbol{a}_j \boldsymbol{u}_j - \sum_{j=1}^{k} \boldsymbol{a}_j \boldsymbol{u}_j = \sum_{j=k+1}^{N} \boldsymbol{a}_j \boldsymbol{u}_j. \tag{A.21}$$

The most commonly used error measure is the squared reconstruction error, which is defined as a square of the length of the residuum. Considering the orthonormality of the eigenvectors $\boldsymbol{u}_j$ we obtain

$$e = \|\boldsymbol{e}\|^2 = \| \sum_{j=k+1}^{N} \boldsymbol{a}_j \boldsymbol{u}_j \|^2 = \sum_{j=k+1}^{N} \boldsymbol{a}_j^2 \tag{A.22}$$

Thus, the squared reconstruction error is equal to the sum of squared discarded principal components. Since they are usually not known, the expected error can be approximated with expected values of the variance across the discarded eigenvectors, which are equal to the corresponding eigenvalues:

$$\varepsilon(\boldsymbol{e}) = \sum_{j=k+1}^{N} \lambda_j \tag{A.23}$$

The expected error is thus equal to the sum of the discarded eigenvalues. This consideration confirms the fact that by maximizing the variance in the first (nondiscarded) eigenvectors, the squared reconstruction error is being simultaneously minimized. These two assertions are indeed two main properties of PCA.

Therefore, for a given dimension of a subspace k, PCA finds such principal vectors $\boldsymbol{u}_l$, $l = 1...k$ and coefficient vectors $\boldsymbol{a}_j \in \mathbb{R}^k$, $j = 1...N$ that minimize the total squared reconstruction error

$$e = \sum_{i=1}^{M} \sum_{j=1}^{N} (\hat{\boldsymbol{x}}_{ij} - \sum_{l=1}^{k} \boldsymbol{u}_{il} \boldsymbol{a}_{lj})^2. \tag{A.24}$$

Thus, as an alternative to the maximization of the variance, the principal vectors and the principal components can be estimated by minimizing the squared reconstruction error.

Appendix B

Expectation-Maximization for Gaussian mixture models

Expectation-Maximization is an efficient iterative algorithm most often used to optimize the parameters of a mixture of Gaussians. A classical paper on Expectation-Maximization is Dempster *et al.* [27]. McLachlan and Krishnan cover Expectation-Maximization and extensions in a book [55].

B.1 Basics of Expectation-Maximization

The objective of Expectation-Maximization (EM) algorithm is to maximize the likelihood $p(\boldsymbol{X};\Theta)$ of the data $\boldsymbol{X}$ drawn from an unknown distribution, given the model parameterized by Θ:

$$\Theta^* = \arg\max_{\Theta} p(\boldsymbol{X}|\Theta) = \arg\max_{\Theta} \prod_{p=1}^{n} p(\boldsymbol{x_p}|\Theta) \tag{B.1}$$

The basic ideas of EM are:

- Introduce a hidden variable such that its knowledge would simplify the maximization of $p(\boldsymbol{X};\Theta)$.
- At each iteration of the algorithm:
  - E-Step: **estimate** the distribution of the hidden variable given the data and the current value of the parameters
  - M-Step: modify the parameters in order to **maximize** the joint distribution of the data and the hidden variable

B.2 Derivation of EM for diagonal Gaussian mixture models

An EM algorithm for Gaussian mixture models can be derived by first assigning the hidden variable a value that estimates for each point which Gaussian generated it.

The EM algorithm can be outlined as follows:

- E-Step: for each point, estimate the probability of each Gaussian generating it;
- M-Step: modify the parameters according to the hidden variable to maximize the likelihood of the data (and the hidden variable).

Let us now derive the EM algorithm for diagonal Gaussian mixture models formally.

Let us call the hidden variable Q. Let us consider the following auxiliary function:

$$A(\Theta; \Theta^s) = E_Q[\log p(\boldsymbol{X}; Q|\Theta)|\boldsymbol{X}; \Theta^s] \tag{B.2}$$

It can be shown that maximizing A

$$\Theta^{s+1} = \arg\max_{\Theta} A(\Theta; \Theta^s) \tag{B.3}$$

always increases the likelihood of the data

$$p(\boldsymbol{X}|\Theta^{s+1}), \tag{B.4}$$

and a maximum of A corresponds to a maximum of the likelihood.

First let us develop the auxiliary function:

$$\begin{aligned}
A(\Theta, \Theta^s) &= E_Q[\log p(\boldsymbol{X}; Q|\Theta)|\boldsymbol{X}; \Theta^s] = & \text{(B.5)}\\
&= \sum_Q P(Q|\boldsymbol{X}; \Theta^s) \log p(\boldsymbol{X}; Q|\Theta) = \\
&= \sum_Q P(Q|\boldsymbol{X}; \Theta^s) \log(P(Q|\boldsymbol{X}; \Theta) \cdot p(\boldsymbol{X}|\Theta)) = \\
&= [\sum_Q P(Q|\boldsymbol{X}; \Theta^s) \log P(Q|\boldsymbol{X}; \Theta)] + [\sum_Q P(Q|\boldsymbol{X}; \Theta^s) \log p(\boldsymbol{X}|\Theta)] = \\
&= [\sum_Q P(Q|\boldsymbol{X}; \Theta^s) \log P(Q|\boldsymbol{X}; \Theta)] + \log p(\boldsymbol{X}|\Theta) & \text{(B.6)}
\end{aligned}$$

then if we evaluate it at Θ^s

$$A(\Theta^s, \Theta^s) = [\sum_Q P(Q|\boldsymbol{X}; \Theta^s) \log P(Q|\boldsymbol{X}; \Theta^s)] + \log p(\boldsymbol{X}|\Theta^s) \tag{B.7}$$

the difference between two consecutive log likelihoods of the data can be written as

$$\log p(\boldsymbol{X}|\Theta) - \log p(\boldsymbol{X}|\Theta^s) = \tag{B.8}$$

$$A(\Theta, \Theta^s) - A(\Theta^s, \Theta^s) + \sum_Q P(Q|\boldsymbol{X}, \Theta^s) \log \frac{P(Q|\boldsymbol{X}; \Theta^s)}{P(Q|\boldsymbol{X}; \Theta)} \tag{B.9}$$

hence, since the last part of the equation is a Kullback-Leibler divergence which is always positive or null, if A increases, the log likelihood of the data also increases. Moreover, one can show that when A is at maximum, the likelihood of the data is also at a maximum.

For GMM, the hidden variable Q will describe which Gaussian generated each example. If Q was observed, then it would be simple to maximize the likelihood of the data: we could just use the estimates for the mean and covariance to calculate the parameters for each class distribution. Moreover, we will see that we can easily estimate Q.

Let us first write the mixture of Gaussian model for one $\boldsymbol{x}_i$:

$$p(\boldsymbol{x}_i|\Theta) = \sum_{j=1}^{N} P(j|\Theta) p(\boldsymbol{x}_i|j, \Theta) \tag{B.10}$$

Let us now introduce the following indicator variable:

$$z_{i,j} = \begin{cases} 1 & \text{if Gaussian } j \text{ emitted } \boldsymbol{x}_i \\ 0 & \text{otherwise} \end{cases} \tag{B.11}$$

We can now write the joint likelihood of all the $\boldsymbol{X}$ and Q:

$$p(\boldsymbol{X}, Q|\Theta) = \prod_{i=1}^{n} \prod_{j=1}^{N} P(j|\Theta)^{z_{i,j}} p(\boldsymbol{x}_i|j, \Theta)^{z_{i,j}} \tag{B.12}$$

which in $\log$ gives

$$\log p(\boldsymbol{X}, Q|\Theta) = \sum_{i=1}^{n} \sum_{j=1}^{N} z_{i,j} P(j|\Theta) + z_{i,j} p(\boldsymbol{x}_i|j, \Theta) \tag{B.13}$$

Let us now write the corresponding auxiliary function:

$$A(\Theta, \Theta^s) = E_Q[\log p(\boldsymbol{X}; Q|\Theta)|\boldsymbol{X}; \Theta^s] = \tag{B.14}$$

$$= E_Q[\sum_{i=1}^{n} \sum_{j=1}^{N} z_{i,j} P(j|\Theta) + z_{i,j} p(\boldsymbol{x}_i|j, \Theta)|\boldsymbol{X}, \Theta^s] = \tag{B.15}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{N} E_Q[z_{i,j}|\boldsymbol{X}, \Theta^s] \log P(j|\boldsymbol{x}_i, \Theta) + E[z_{i,j}|\boldsymbol{X}, \Theta^s] \log p(\boldsymbol{x}_i|\Theta)$$

Hence, the E-Step estimates the posterior:

$$
\begin{aligned}
E_Q[z_{i,j}|\boldsymbol{X},\Theta^s] &= 1\cdot P(z_{i,j}=1|\boldsymbol{X},\Theta^s)+0\cdot P(z_{i,j}=0|\boldsymbol{X},\Theta^s)= && \text{(B.16)}\\
&= P(j|\boldsymbol{x}_i,\Theta^s)= && \text{(B.17)}\\
&= \frac{p(\boldsymbol{x}_i|j,\Theta^s)P(j|\Theta^s)}{p(\boldsymbol{x}_i|\Theta^s)} && \text{(B.18)}
\end{aligned}
$$

and the M-step finds the parameters Θ that maximizes A, hence searching for

$$
\frac{\partial A}{\partial \Theta}=0 \tag{B.19}
$$

for each parameter (means $\boldsymbol{\mu}_j$, variances σ_j^2, and weights w_j). Note however that for the weights w_j , we need to enforce their sum to 1 by adding a Lagrange term.

Let us develop $\frac{\partial A}{\partial \Theta}=0$ for $\boldsymbol{\mu}_j$

$$
\begin{aligned}
\frac{\partial A}{\partial \boldsymbol{\mu}_j} &= \sum_{i=1}^{n}\frac{\partial A}{\partial \log p(\boldsymbol{x}_i,\Theta)}\frac{\partial \log p(\boldsymbol{x}_i,\Theta)}{\partial \boldsymbol{\mu}_j} && \text{(B.20)}\\
&= \sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\frac{\partial \log p(\boldsymbol{x}_i,\Theta)}{\partial \boldsymbol{\mu}_j}\\
&= \sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\frac{\partial \log p(\boldsymbol{x}_i,\Theta)}{\partial p(\boldsymbol{x}_i|\Theta)}\frac{\partial p(\boldsymbol{x}_i|\Theta)}{\partial p(\boldsymbol{x}_i|j,\Theta)}\frac{\partial p(\boldsymbol{x}_i|j,\Theta)}{\partial \log p(\boldsymbol{x}_i|j,\Theta)}\frac{\partial \log p(\boldsymbol{x}_i|j,\Theta)}{\partial \boldsymbol{\mu}_j}\\
&= \sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\cdot\frac{1}{p(\boldsymbol{x}_i|\Theta)}\cdot w_j\cdot p(\boldsymbol{x}_i|\Theta)\cdot\frac{\boldsymbol{x}_i-\boldsymbol{\mu}_j}{\sigma_j^2}=0 && \text{(B.21)}
\end{aligned}
$$

By solving above we gain:

$$
\begin{aligned}
\sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\cdot w_j\cdot\boldsymbol{x}_i &- \sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\cdot w_j\cdot\hat{\boldsymbol{\mu}}_j=0 && \text{(B.22)}\\
\frac{\sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\cdot w_j\cdot\boldsymbol{x}_i}{\sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\cdot w_j} &= \hat{\boldsymbol{\mu}}_j && \text{(B.23)}\\
\frac{\sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)\cdot\boldsymbol{x}_i}{\sum_{i=1}^{n}P(j|\boldsymbol{x}_i,\Theta^s)} &= \hat{\boldsymbol{\mu}}_j && \text{(B.24)}
\end{aligned}
$$

By solving for all variables we finally arrive at the end results:

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}_j &= \frac{\sum_{i=1}^{N}\boldsymbol{x}_iP(j|\boldsymbol{x}_i,\Theta^s)}{\sum_{i=1}^{N}P(j|\boldsymbol{x}_i,\Theta^s)} && \text{(B.25)}\\
(\hat{\sigma}_j)^2 &= \frac{\sum_{i=1}^{N}(\boldsymbol{x}_i-\hat{\boldsymbol{\mu}}_j)^2P(j|\boldsymbol{x}_i,\Theta^s)}{\sum_{i=1}^{N}P(j|\boldsymbol{x}_i,\Theta^s)} && \text{(B.26)}\\
\hat{w}_j &= P(j|\Theta)=\frac{1}{N}\sum_{i=1}^{N}P(j|\boldsymbol{x}_i,\Theta^s) && \text{(B.27)}
\end{aligned}
$$

B.3 Initialization

EM is an iterative procedure that is only guaranteed to converge monotonically to one of the local maxima, therefore it is very sensitive to initial conditions! If the initial approximation is bad, the algorithm may not converge to a useful local maximum. Hence, we need a good and fast initialization procedure. Most often used in practice are K-Means clustering and random initialization, which can be facilitated by initializing variances based on the data. Other options include hierarchical K-Means and Gaussian splitting. Convergence properties of the EM algorithm for Gaussian mixtures were studied by Xu and Jordan [97].

Dodatek C

Razširjen povzetek v slovenščini

C.1 Uvod

Na področju računalniškega vida je bilo razvitih že več metod za sledenje in identifikacijo človeške hoje na podlagi izgleda ([47, 63, 103, 49, 98, 9, 29, 69, 43, 100, 71, 51, 28, 60, 61, 53, 54, 102, 38, 72, 5, 20, 33, 25, 32, 101] itd.). Pristope lahko v grobem razdelimo po tem, ali izhajajo iz modela ali iz lokalnih značilnic. Pristopi, ki izhajajo iz modela, tipično privzamejo prostorski model in ocenjujejo razvoj konfiguracijskih parametrov skozi čas. Pristopi, ki izhajajo iz lokalnih značilnic, skušajo zgraditi prostorski model iz množice primitivov in nadaljujejo z oceno razvoja skozi čas.

Prostorski modeli, ki jih najdemo v literaturi, so v splošnem zasnovani na slikah [5, 21, 9, 32] ali na geometriji, ki je lahko 2-dimenzionalna [60, 20, 102] ali 3-dimenzionalna [64, 103, 77, 29]. Zloženi modeli [49] sestavljajo geometrično strukturo iz 2-dimenzionalnih slikovnih predlog. Za razpoznavanje načina hoje številni avtorji [60, 61, 53, 25, 102, 20, 101] zgolj sledijo nekaterim vnaprej predpostavljenim značilnicam in ne modelirajo celotnega vidnega polja. Bilo je že nekaj poskusov uporabe slikovnih podprostorov [32, 9] in slikovnih statistik [72], da bi zmanjšali dimenzionalnost prostorske predstavitve, vendar so te metode nagnjene k izgubi opisne moči za posamezne lokalne okolice. Večina metod je omejenih pri predstavitvi objektov, ki odstopajo od predpostavk o prostorski konfiguraciji.

Časovni razvoj je tipično predstavljen s časovnim zaporedjem parametrov ali s prehodi med stanji [103], običajno modelirani s prikritimi Markovimi modeli [51]. Metode zasnovane na stanjih lahko enostavno predstavijo številna različna gibanja, vendar niso posebej primerne za predstavitev podrobnosti gibov, ker bi potrebovali preveč stanj za modeliranje vseh mogočih konfiguracij in lokalnih variacij.

C.2 Naš pristop

Na področju analize človeške hoje je bilo opravljeno veliko dela predvsem na podatkih pridobljenih z zajemanjem gibanja označb pripetih na človeške igralce. Cedras in Shah [16, 17] opisujeta eksperimente Johanssona [47] in ostalih, ki kažejo, da ljudje uspešno prepoznajo človeško gibanje z opazovanjem zelo majhne množice označb (celo v prisotnosti šuma).

Bilo pa je malo poskusov učenja človeške hoje na podlagi gibanja lokalnih značilnic brez uporabe označb. Niyogi in Adelson [60, 61] sta uporabljala prostorsko-časovno mnogoterost, pridobljeno z razvojem roba človeškega obrisa skozi čas, vendar nista modelirala niti gibanja znotraj obrisa niti gibanja v navpični smeri. Uvedla sta idejo kanoničnega gibanja, vendar ne podajata načina za posplošitev ali učenje takega modela z upoštevanjem verjetnosti. Torresani in sod. [85] se poskušajo naučiti gibanja površine iz video posnetka, vendar uporabljajo označene podatke in stremijo k učenju natančne oblike, mi pa skušamo uporabiti neoznačene podatke, da bi se naučili sledi bolj zapletene strukturirane površine v gibanju.

Kot nam je znano, do sedaj ni nihče poskušal modelirati človeške hoje kot prostorsko-časovno mnogoterost celotne vidne površine. Glavna prednost takega modela je zmožnost verjetnostne predstavitve strukture in gibanja v skupnem ogrodju, z možnostjo vključevanja lokalnih in globalnih variacij.

Izhajamo iz predpostavke, da je možno izgled gibanja sestavljenega objekta zadostno predstaviti z množico trajektorij točk na površini objekta, če je število sledenih točk zadostno za približen opis gibajočih površin in prostostnih stopenj. Osredotočamo se na ciklično človeško gibanje, ker je na voljo več baz posnetkov in ker nam dodatne fizikalne omejitve omogočajo, da uporabimo preprost postopek za sledenje točkam in izločanje neoptimalnih trajektorij.

Predstavljamo novo metodo za predstavitev strukturiranega cikličnega gibanja, zasnovno na množici prostorsko-časovnih trajektorij zvezno sledenih točk na površini opazovanega objekta. Metodo uporabimo za vizualno učenje in razpoznavanje človeške hoje. Ne predpostavljamo predhodne informacije o razporeditvi trajektorij. Glavna prednost metode je, da verjetnostno modelira gibanje čez celotno vidno polje in skozi čas. Trenutno metoda uporablja samo zvezno sledljive površinske točke, vendar lahko v principu v model vključimo kakršne koli sledljive značilnice.

Glavni prispevek magistrske naloge je metoda za vizualno učenje in razpoznavanje prostorsko-časovne porazdelitve množice prostorsko-časovnih krivulj preko večih ponovitev cikličnega gibanja. Prostorsko-časovne krivulje posplošimo skozi čas s

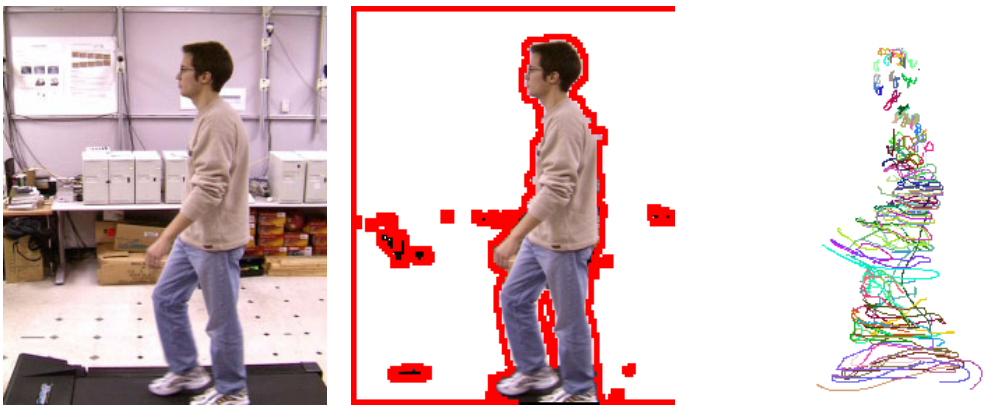
pomočjo analize glavnih komponent. Porazdelitev izrazimo kot približek z uporabo mešanice Gaussov. Razpoznavanje je izvedeno s kombinacijo maksimalne aposteriorne verjetnosti in linearnega prilagajanja podatkov.

C.3 Sledenje točk na gibajočem objektu v posnetku

Skupni predhodnik za učenje in razpoznavanje iz video posnetka gibanja je izsejanje prostorsko-časovnih krivulj. S sledenjem naključnih točk na površini opazovanih ljudi skušamo pridobiti množico prostorsko-časovnih krivulj.

Predpostavljamo, da gibanje opazujemo v pol-kontroliranem okolju, kar nam pomeni okolje s pretežno statičnim ozadjem in stalno osvetlitvijo. Kvaliteta segmentacije je kritična predvsem za učno fazo, ker skušamo izločiti vpliv odstopajočih podatkov in se naučiti le pomembnih značilnic, ki zares pripadajo opazovani osebi. Predlagana metoda razpoznavanja pa lahko obravnava odstopajoče podatke probabilistično.

Implementirali smo standardni postopek izločanja gibajočega objekta, ki je sestavljen iz normalizacije osvetlitve, odštevanja ozadja, odstranjevanja senc in filtriranja šuma. S tem, da omejimo sledenje točk na notranjost segmentirane površine gibajočega objekta, si olajšamo predvsem filtriranje nepomembnih podatkov o ozadju in izognemo sledenju težko sledljivih točk na robovih udov.



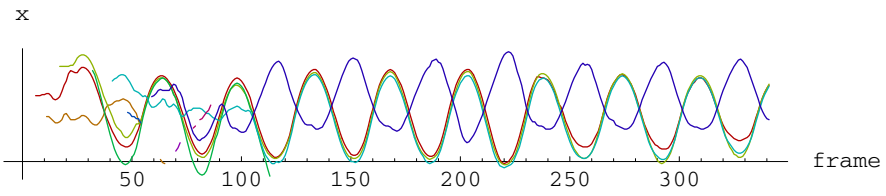
Slika C.1: Originalna slika, izločanje gibajočega objekta in sledenje točk

Namesto, da bi skušali slediti vsem točkam na površini, predlagamo naključno vzorčenje trajektorij točk na podlagi optičnega toka majhnega bloka okrog sledene točke.

Implementirali smo postopek, ki skuša vzorčiti površino objekta enakomerno v prostoru in času. Do neke mere tolerira začasno izgubljene dele s predikcijo hitrosti in

ponovno vzorči dele, ki se na novo pojavijo. Lahko določimo zgornjo mejo za gostoto vzorčenja.

Za vsako točko dobimo časovno zaporedje, ki opisuje trajektorijo s koordinatami $\boldsymbol{c}(t) = (c_x(t), c_y(t))$ v vidnem polju (view-space) in oznako prisotnosti $q(t)$, ki označuje, če smo točko opazili na sliki v trenutku t. Trajektorija ni nujno zvezna in včasih napak skače med deli objekta (glej sliko C.2).



Slika C.2: Primeri trajektorij pridobljenih s sledenjem 10 naključnih točk

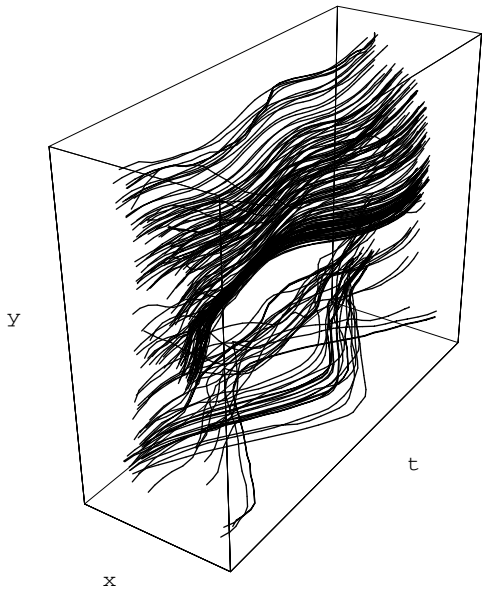
C.4 Detekcija ciklov in izsejanje krivulj

En cikel definiramo kot interval, ki vsebuje dva človeška koraka. Cikle v kratkih posnetkih (okoli 3–10 ciklov) hoje detektiramo z iskanjem maksimumov avtokorelacij trajektorij in glasovanjem.

Detekcijo ciklov kontroliramo s parametroma C_{min} in C_{max}, ki določata minimalno in maksimalno dolžino cikla. Za vsako trajektorijo določimo referenčno okno dolžine C_{max} na sredini posnetka in izračunamo avtokorelacijo z vsemi drugimi okni na trajektoriji povsod, kjer je trajektorija zvezna skozi obe okni. Če je lokalni maksimum v zaporedni sliki f tudi maksimum na intervalu $[f - C_{min}, f + C_{min}]$, dodamo 1 glas v ustrezni element glasovalnega akumulacijskega vektorja $\boldsymbol{b}_f$.

Glasovalni vektor $\boldsymbol{b}$ nato zgladimo z jedrom $(1, 2, 1)$. Lokalni maksimumi f, kjer je $\boldsymbol{b}_f = \max_{i=f-C_{min}}^{f+C_{min}} \boldsymbol{b}_i$, določajo kandidate za začetek cikla. Zaporedne kandidate odštejemo, da dobimo seznam možnih dolžin cikla. Mediana dolžin ciklov je izbrana kot končna ocena za dolžino cikla C. Začetne slike kandidatov, ki padejo v intervale $k * C \pm \frac{C}{4}$ so izbrane kot končni začetki ciklov.

Izsejemo samo povezane dele trajektorij, ki se pričenjajo na mestih detektiranih ciklov. Rezultat je množica fazno poravnanih, skoraj cikličnih prostorsko-časovnih krivulj (glej sliko C.3) predstavljenim s časovnim zaporedjem 2-dimenzionalnih koordinat. Krivulje popravimo do ciklične oblike z linearnim popravkom, nato jih odsekoma linearno interpoliramo na skupno dolžino L. Središče krivulje odštejemo od koordinat točk. Končna



Slika C.3: Množica fazno poravnanih prostorsko-časovnih krivulj

predstavitev prostorsko-časovne krivulje sestoji iz središča krivulje $\boldsymbol{o} = (o_x, o_y)$ in vektorja oblike krivulje $\boldsymbol{x} = [x_1, y_1, ..., x_L, y_L]$, ki mu je središče že odšteto.

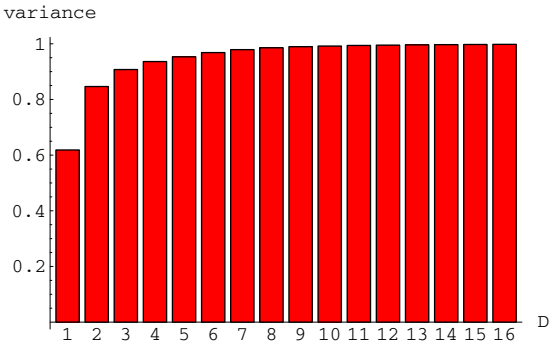
C.5 Probabilistični prostorsko-časovni model

V tem razdelku je opisano učenje verjetnostnega prostorsko-časovnega modela množice prostorsko-časovnih krivulj. Postopek učenja je razdeljen na dekompozicijo vektorjev krivulj v prostoru glavnih komponent in modeliranje porazdelitve krivulj v podprostoru s pomočjo mešanice Gaussov.

C.5.1 Množica krivulj v podprostoru glavnih komponent

Naj bo $\boldsymbol{X}$ matrika podatkov z N vektorji oblike krivulj $\boldsymbol{x}_n$ dolžine $D = 2 * L$ v urejenih stolpcih. Dekompozicijo v prostoru glavnih komponent izvedemo po postopku Andersona [3].

$$\begin{aligned} \boldsymbol{\mu} &= [\mu_1, ..., \mu_D]^T = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \\ \hat{\boldsymbol{X}} &= \boldsymbol{X} - \boldsymbol{\mu} 1_{1\times N} \\ \boldsymbol{C} &= \frac{1}{N}\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^T \end{aligned} \tag{C.1}$$



Slika C.4: Varianca vsebovana v prvih D lastnih vektorjih

Kovariančno matriko $\boldsymbol{C}$ lahko po postopku razcepa z glavnimi vrednostmi diagonaliziramo

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{C.2}$$

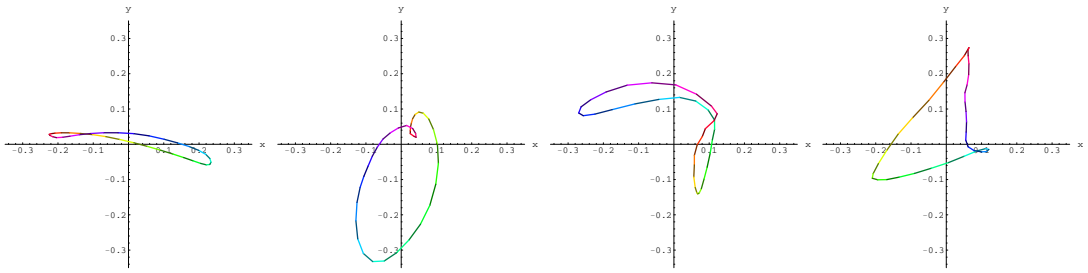
na tak način, da ortonormalna matrika $\boldsymbol{U}$ vsebuje glavne vektorje $[\boldsymbol{u}_1, ..., \boldsymbol{u}_D]$ v stolpcih in diagonalna matrika $\boldsymbol{\Lambda}$ vsebuje lastne vrednosti λ_i na diagonali ter da so lastne vrednosti in ustrezni lastni vektorji urejeni v padajočem redu lastnih vrednosti. Tako je večina variabilnosti množice krivulj vsebovana že v prvih nekaj lastnih vektorjih, tako imenovanih glavnih vektorjih. Matriko $\boldsymbol{U}$ uporabimo, da preslikamo vektorje oblike krivulj $\hat{\boldsymbol{X}}$ na glavne osi:

$$\boldsymbol{P} = \boldsymbol{U}^T\hat{\boldsymbol{X}} \tag{C.3}$$

Lastnosti preslikave nam zagotavljajo, da z zmanjšanjem predstavitve vektorjev oblike krivulj na prvih nekaj glavnih komponent minimiziramo rekonstrukcijsko napako po merilu povprečne kvadratične napake. Predstavitev krivulj z uporabo podprostora glavnih krivulj za predstavitev prostorsko-časovne variacije postane $\boldsymbol{r} = [o_x, o_y, p_1, ..., p_D]$ (koordinate središča ohranimo nespremenjene).

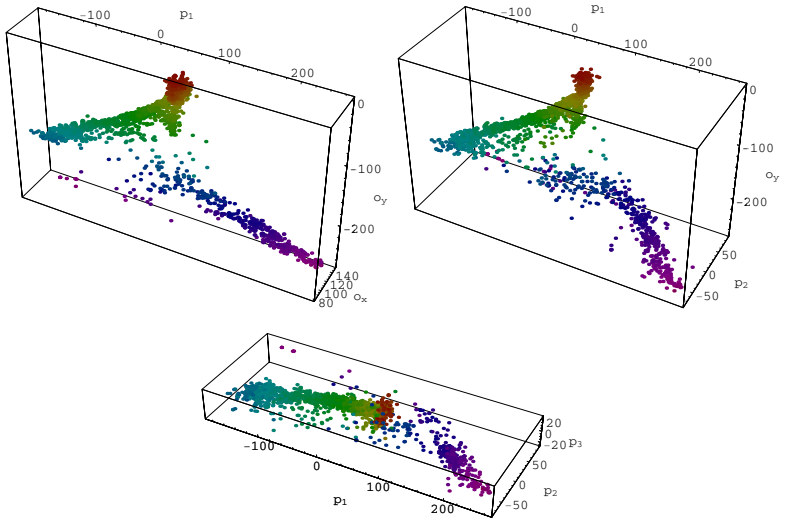
Analizirali smo diagrame glavnih vektorjev. V vseh primerih stranskega pogleda je prvi glavni vektor skoraj cikličen in vsebuje izrazito nihanje vzdolž smeri gibanja (glej sliko C.5). To lastnost v nadaljevanju izkoriščamo za fazno poravnavanje krivulj.

Diagrami porazdelitve množic krivulj v podprostoru (glej sliko C.6 levo) nekoliko spominjajo na palične figure, ki so pogosto uporabljene kot enostaven model za sledenje in analizo načina gibanja. Tu je bistvena razlika v tem, da glavna horizontalna komponenta predstavlja prvi glavni vektor prostorsko-časovne krivulje namesto prostorskega odmika, vključeni pa so tudi nekateri osamelci, ki so posledica napak pri sledenju točk.



Slika C.5: Prvi 4 glavni vektorji (od leve proti desni)

Dva uda manjkata zaradi zakrivanj in ker z našim algoritmom niti ne poskušamo obravnavati nepovezanih trajektorij.



Slika C.6: Preslikave množice prostorsko-časovnih krivulj v podprostore

C.5.2 Modeliranje porazdelitve krivulj z mešanico Gaussov

Gostoto porazdelitve prostorsko-časovnih krivulj predstavimo kot približek z mešanico Gaussov Θ.

$$\mathcal{N}(\boldsymbol{r};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{|\boldsymbol{r}|}{2}}\sqrt{|\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{r}-\boldsymbol{\mu})) \tag{C.4}$$

$$p(\boldsymbol{r}) = \sum_{i=1}^{K} w_i * \mathcal{N}(\boldsymbol{r};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \tag{C.5}$$

K je število Gaussov in w_i je utež Gaussa i, kjer je $\sum_{i=1}^{K} w_i = 1$ in $\forall i : w_i \geq 0$. Uporabljamo diagonalne $\mathbf{\Sigma}_i$ z vrednostmi $\sigma_1 ... \sigma_K$ na diagonali.

Srednje vrednosti Gaussov inicializiramo tako, da jih nastavimo na naključno podmnožico vektorjev podatkov. Začetno varianco nastavimo na naključni del intervala, ki ga zasedajo podatki.

Za optimizacijo modela uporabimo iterativni postopek Expectation-Maximization [27]:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_j &= \frac{\sum_{i=1}^{N} \boldsymbol{r}_i P(j|\boldsymbol{r}_i, \Theta^s)}{\sum_{i=1}^{N} P(j|\boldsymbol{r}_i, \Theta^s)} \\ (\hat{\sigma}_j)^2 &= \frac{\sum_{i=1}^{N} (\boldsymbol{r}_i - \hat{\boldsymbol{\mu}}_j)^2 P(j|\boldsymbol{r}_i, \Theta^s)}{\sum_{i=1}^{N} P(j|\boldsymbol{r}_i, \Theta^s)} \\ \hat{w}_j &= P(j|\Theta) = \frac{1}{N} \sum_{i=1}^{N} P(j|\boldsymbol{r}_i, \Theta^s) \end{aligned} \tag{C.6}$$

Postopek Expectation-Maximization lokalno optimizira parametere Θ in z vsakim korakom monotono povečuje verjetnost, vendar ni zagotovljeno, da najdemo globalni maksimum. Zato poženemo postopek večkrat z različnimi začetnimi vrednostmi in izberemo rezultat, ki maksimizira logaritem matematičnega upanja:

$$\log p(\boldsymbol{r}_1 ... \boldsymbol{r}_N|\Theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} w_k * \mathcal{N}(\boldsymbol{r}_i; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k) \; . \tag{C.7}$$

Končni model opazovanega gibanja je tako sestavljen iz glavnih vektorjev $u_1 ... u_D$, ki modelirajo prostorsko-časovno variacijo trajektroij in množico parametrov Gaussov $\{w_i, \mu_i, \sigma_i\}$, $i = 1...K$, ki modelirajo gostoto porazdelitve trajektorij v kombiniranem 2-dimenzionalnem vidnem prostoru in D-dimenzionalnem podprostoru oblike prostorsko-časovnih krivulj.

C.6 Razpoznavanje

Postopek za razpoznavanje je enak kot za učenje do razcepa na glavne komponente. Od tega koraka nadaljujemo s prostorsko-časovnim poravnavanjem in oceno maksimalne aposteriorne verjetnosti za klasifikacijo.

C.6.1 Prostorsko-časovno poravnavanje

Dano množico novih opaženih trajektorij najprej časovno poravnamo, tako da izračunamo prvi glavni vektor in izberemo fazo, ki maksimizira korelacijo s prvim glavnim vektorjem predhodno naučenega modela. Obravnavati moramo tako originalni kot negirani prvi

glavni vektor, kar nam da dva možna rezultata za fazni zamik, ker ne moremo predpostaviti orientacije lastnih vektorjev po postopku razcepa.

Predpostavljamo, da lahko približno prostorsko poravnavo dosežemo z drugimi postopki, torej lahko uporabimo izčrpno preiskovanje na relativno majhnem območju zanimanja, kar izvedemo z linearnim prilagajanjem podatkovnih vektorjev.

C.6.2 Klasifikacija

Pričnemo s prostorsko-časovno poravnanimi trajektorijami iz prejšnjega koraka in preračunamo novo podatkovno matriko opazovanih vektorjev $\boldsymbol{Y} = [\boldsymbol{y}_1 ... \boldsymbol{y}_N]$ preslikanih v podprostor z glavnimi osmi vsakega predhodno naučenega modela c.

Želimo najti model c_i, ki maksimizira maksimalno aposteriorno verjetnost $p(c_i|\boldsymbol{Y})$ ob dani množici vektorjev krivulj $\boldsymbol{Y}$. Uporabimo Bayesovo pravilo:

$$\arg\max_i p(c_i|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|c_i)p(c_i)}{p(\boldsymbol{Y})} \tag{C.8}$$

S predpostavko vnaprej enako verjetnih modelov c_i in z opazko, da je $p(\boldsymbol{Y})$ enako za vse modele, se klasifikacija poenostavi na

$$\arg\max_i p(\boldsymbol{Y}|c_i) \tag{C.9}$$

Ker je opazovanje posameznih trajektorij med seboj neodvisno, je končno klasifikacijsko pravilo poenostavljeno na:

$$\arg\max_i \prod_{n=0}^{N} p(\boldsymbol{y}_n|c_i) = \arg\max_i \sum_{n=0}^{N} \log p(\boldsymbol{y}_n|c_i). \tag{C.10}$$

Model c_i, ki maksimizira (C.10), je izbran kot najbolj verjeten kandidat.

Izračunavanje $p(\boldsymbol{y}_n|c_i)$ za razpoznavanje zahteva, da preslikamo $\boldsymbol{y}_n$ v glavni podprostor c_i, kar praktično zahteva prostorsko-časovno preslikavo matrike podatkov $\boldsymbol{Y}$ in izračun $\sum_{n=0}^{N} \log p(\boldsymbol{y}_n|c_i)$ za vsak razred posebej. Za namene sledenja pa moramo zgolj izračunati aposteriorno verjetnost posameznega razreda.

C.7 Eksperimenti

Za večino poskusov uporabljamo bazo CMU MoBo [41]. To je baza kratkih video posnetkov 25 ljudi, ki izvajajo 4 načine hoje na tekočem traku in so posneti iz 6 zornih kotov istočano. Posnetki vsebujejo 300-340 slik.

Osredotočili smo se predvsem na posnetke hoje iz stranskega pogleda, ker vsebujejo največ prostorske dinamike, ki je potencialno uporabna za razpoznavanje.

Nekaj testov smo naredili tudi na lastnih posnetkih v nenadzorovanem okolju z namenom, da bi ugotovili kakšne podprobleme je potrebno rešiti v nadaljnem delu.

C.7.1 Učenje

Po opisanih postopkih smo obdelali video posnetke in analizirali njihove lastne vektorje (glej sliko 3.9) in količino variance vsebovane v prvih nekaj glavnih vektorjih. Opazili smo, da prvi glavni vektor vsebuje 38.2%–86.6% variance, prvi 4 vektorji vsebujejo 84.7%–96.9% variance in prvih 16 vektorjev vsebuje že več kot 99.4% variance.

Izvedli smo sledenje točk, odkrivanje ciklov in izločanje krivulj. Dolžino cikla smo omejili med 20 in 50 slik. Opazili smo, da ocenjena dolžina cikla C variira med 28 in 39 posnetki. Krivulje smo odsekoma linearno interpolirali na enotno dolžino, ki smo jo določili kot $L = 32$ točk. Posnetke smo razdelili na pol, zadnjo polovico smo rabili za učenje in prvo polovico za testiranje. Za predstavitev prostorsko-časovnih variacij smo ohranili samo 4 glavne komponente. Podatkovni vektorji so tako vsebovali 2 prostorska parametra in 4 prostorsko-časovne parametre. V postopku učenja smo gradili diagonalni model mešanih Gaussov s 15 Gaussi po postopku EM inicializiranem na naključni podmnožici podatkov, z naključno varianco skalirano iz intervala podatkov.

Število iteracij postopka EM smo določili s poskusi. Izkazalo se je, da konvergira v večini primerov po okrog 30 iteracijah. Za teste smo uporabili 30 iteracij EM po 15 naključnih inicializacijah in ohranili najboljši model, ki je maksimiziral matematično upanje za množico učnih vektorjev.

Število Gaussov smo najprej določili empirično z opazovanjem animacij trajektorij, ki jih predstavljajo srednje vrednosti Gaussov. Nato smo preverili klasifikacijo z variiranjem števila Gaussov. Izkazalo se je, da metoda ni zelo občutljiva na število Gaussov, dokaj dobro delovanje s samo 5 Gaussi pa celo kaže na to, da je oblika krivulj bistveno bolj pomembna, kot točen opis porazdelitve.

C.7.2 Razpoznavanje

Vektorje posnetkov smo fazno poravnali z maksimiziranjem korelacije prvih glavnih vektorjev dveh zaporedij. Za hitro hojo smo preverili natančnost faznega poravnavanja in nismo odkrili večjih odstopanj kot eno kvantizacijsko enoto. Opazovane vektorje smo preslikali v predhodno naučeni podprostor z uporabo le prvih 4 glavnih komponent. Te-

način hoje	pogled	št. posnetkov	št. napak prepoznave	zaporedno mesto pravilnega razreda
hitra hoja	stranski	25	1	drugi
počasna hoja	stranski	25	0	
hoja vkreber	stranski	25	1	drugi
hoja z žogo	stranski	24	0	
počasna hoja	od zadaj $45°$	25	1	tretji
vsi	stranski	99	2	

Tabela C.1: Povzetek rezultatov razpoznavanja

stirali smo na različnih prostorskih odmikih z izvajanjem izčrpnega preiskovanja v rangu $[-32...32]$ za x in $[-16...16]$ za y os z velikostjo koraka 4.

Za razpoznavanje identitete smo klasificirali po 25 testnih posnetkov z istim načinom gibanja proti 25 učnim posnetkom z oceno maksimalne aposteriorne verjetnosti (pri hoji z žogo je v bazi manjkal en posnetek). Za razpoznavanje identitete in akcije istočasno smo klasificirali posnetke iz stranskega pogleda vse proti vsem.

Povzetek rezultatov testov je v tabeli C.1. Pregledali smo napačno klasificirane posnetke. Opazili smo, da so najbolj verjetni razlogi za napačno klasificiranje necikl̆ične kretnje roke, veliki odmiki položaja v prostoru in variacija dolžine cikla znotraj enega posnetka, ki je naš algoritem ne upošteva. V nekaterih poskusih so k napačni klasifikaciji prispevali tudi slabi Gaussovi modeli, ki pa smo jih lahko izboljšali s ponovnimi poskusi učenja z večimi iteracijami.

C.7.3 Testi v nenadzorovanem okolju

S testi v nenadzorovanem okolju smo ugotovili, da bi morali za neposredno uporabo predlagane metode predhodno uporabiti postopke, ki bi odstraniti translacijo in ocenili velikost opazovanega objekta. Na podlagi velikosti bi lahko tudi optimizirali način in gostoto sledenja točk.

Če pa bi želeli naš pristop posplošiti, bi morali razrešiti odvisnost od zornega kota, upoštevati variacije cikla, izločiti verjetne necikl̆iče kretnje, ugotavljati prehode med načini in smermi gibanja, kombinirati sledenje in razpoznavanje, ter nazadnje slediti in ločevati več oseb v interakcijah.

C.8 Razprava

Rezultati potrjujejo izhodiščno idejo, da je možno človeško hojo avtomatično in učinkovito modelirati, se je učiti in jo razpoznati z uporabo modela mešanih Gaussov za predstavitev gostote porazdelitve prostorsko-časovnih krivulj v avtomatsko naučenem optimalnem podprostoru.

Vendar na podlagi našega razumevanja metode ugotavljamo potrebo za nadaljne testiranje, da bi izločili vpliv velikosti, obleke, teksturne odvisnosti in odvisnosti od dolžine cikla, preden bi metodo priporočili kot splošen pristop k prepoznavanju načina gibanja. Menimo, da bi se napake v oceni velikosti in koordinatnega izhodišča lahko izkazale kot kritične za praktičen uspeh.

Metoda je odvisna od zornega kota, vendar bi lahko teoretično vključili nekaj variacij v predhodno naučen verjetnostni model.

Metoda nudi možnost za prenos znanja o gibanju iz laboratorija v nenadzorovano okolje. Prenos znanja iz osebe na osebo je možen, vkolikor sklepamo iz podobnosti podmnožice videnih oziroma naučenih gibov na podobnost nevidenih gibov. Možno je tudi zgraditi pramodele za razrede velikosti, postav in načinov gibanja. Prenos znanja iz aktivnosti na aktivnost je trši oreh, kajti potrebno je ugotoviti korespondence med središči Gaussov in modificirati podmnožico.

Neposredne izboljšave metode so možne predvsem z drugačnim načinom modeliranja in primerjanja porazdelitve. Dvosmerno primerjanje porazdelitve bi odpravilo težavo z neupoštevanjem delov predhodne porazdelitve, kjer sploh ni testnih podatkov. Učenje bi lahko napravili deterministično npr. z grupiranjem podatkov in opisom z radialnimi baznimi funkcijami.

Zahtevo po zvezno sledljivih trajektorijah bi lahko odpravili z dopolnjevanjem manjkajočih podatkov ali z uporabo robustnih postopkov za učenje podprostorov [13].

Razširitev v 3D prostor pa je preprosta, če uporabimo 3D sledenje, saj dodatna koordinata v vektorjih v ničemer ne spremeni opisanih postopkov. Ta način bi celo priporočili za učenje pramodelov, ki bi jih za razpoznavanje preslikali v 2D podprostor določen z zornim kotom kamere.

C.9 Zaključek

Predlagali smo nov prostorsko-časovni model za predstavitev cikličnega človeškega gibanja v vidnem polju ene kamere, skupaj s postopki za učenje in razpoznavanje.

Rezultati na bazi CMU MoBo za prepoznavanje ljudi na podlagi hoje so vzpodbudni.

Na podlagi rezultatov trdimo, da je verjetnostno modeliranje hoje na podlagi lokalnih prostorsko-časovnih trajektorij zanimiv pristop k učenju in razpoznavanju modelov hoje za razpoznavanje in sledenje.

Testi v nenadzorovanem okolju kažejo še na številne nerešene podprobleme, ki jih je potrebno rešiti, da bi lahko neposredno uporabili ali posplošili postopke.

Razviti postopki ne zahtevajo označb in ne predpostavljajo specifične strukture, zato bi jih lahko uporabili tudi za učenje in razpoznavanje drugih tipov cikličnega gibanja. Kot nam je znano, smo se prvi uspeli naučiti in prepoznati tak model zgolj na podlagi lokalnih prostorsko-časovnih značilnic.

Naš pristop k predstavitvi množice prostorsko-časovnih krivulj lahko vidimo tudi kot novo prostorsko posplošitev posamičnih glavnih krivulj. Postopek je splošen in bi ga lahko uporabili za učenje drugih množic prostorsko-časovnih krivulj.

Bibliography

[1] M. Afify and O. Siohan. Constrained maximum likelihood linear regression for speaker adaptation. In *Proc. Int. Conf. on Spoken Language Processing*, 2000.

[2] Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. Discovering Clusters in Motion Time-Series Data. In *Proc. IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, June 2003.

[3] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 1958.

[4] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance Of Optical Flow Techniques. *CVPR*, pages 236–242, 1992.

[5] Chiraz BenAbdelkader, Ross Cutler, Harsh Nanda, and L. S. Davis. EigenGait: Motion-based Recognition of People using Image Self-similarity. In *Proc. Intl Conf. on Audio and Video-based Person Authentication (AVBPA)*, 2001.

[6] Samy Bengio. An Introduction to Statistical Machine Learning - EM for GMMs. http://www.idiap.ch/ bengio/lectures/tex_gmm.pdf, 2003. IDIAP Statistical Machine Learning Lectures.

[7] R. J. V. Bertin and I. Israel. Optic flow based perception of two-dimensional trajectories and the effects of a single landmark. http://cogprints.ecs.soton.ac.uk/archive/00002830/.

[8] Neil Birkbeck. Structure and Motion Algorithms. http://www.cs.ualberta.ca/ birkbeck/c603presentation.ppt, November 2003. Course Project C603 presentation.

[9] M.J. Black, Y. Yacoob, and X. S. Ju. Recognizing human motion using parameterized models of optical flow. In Mubarak Shah and Ramesh Jain, editors, *Motion-Based Recognition*, pages 245–269. Kluwer Academic Publishers, 1997.

[10] Robert Bodor, Bennett Jackson, Osama Masoud, and Nikolaos Papanikolopoulos. Image-Based Reconstruction for View-Independent Human Motion Recognition. In *Proc. of the IEEE/RJS Intl. Conf. on Intelligent Robots and Systems*, October 2003.

[11] Robert Bodor, Bennett Jackson, and Nikolaos Papanikolopoulos. Vision-Based Human Tracking and Activity Recognition. In *Proc. of the 11th Mediterranean Conf. on Control and Automation*, June 2003.

[12] Constantinos Boulis. Adaptation techniques for speaker recognition. citeseer.ist.psu.edu/boulis02adaptation.html, 2002.

[13] D. Skočaj. *Robust Subspace Approaches to Visual Learning and Recognition*. PhD thesis, Faculty of Computer and Information Science, University of Ljubljana, 2003.

[14] Yaron Caspi and Michal Irani. A Step Towards Sequence-to-Sequence Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–689, June 2000.

[15] Yaron Caspi and Michal Irani. Spatio-Temporal Alignment of Sequences. *PAMI*, 24(11):1409–1424, November 2002.

[16] C. Cedras and M. Shah. A Survey of Motion Analysis from Moving Light Displays. In *Proc. 1994 IEEE Conf. on Computer Vision and Pattern Rec.*, pages 214–221. IEEE Press, 1994.

[17] C. Cedras and M. Shah. Motion-Based Recognition: A Survey. *IVC*, 13(2):129–155, March 1995.

[18] Dmitry Chetverikov and Judit Verestoy. Tracking Feature Points: a New Algorithm. In *Proc. of 14th International Conference on Pattern Recognition*, pages 1436–1438, 1998.

[19] Denis V. Chigirev and William S. Bialek. Optimal Manifold Representation of Data: An Information Theoretic Approach. In *Proc. Of Neural Information Processing Systems (NIPS)*, 2003.

[20] Robert T. Collins, Ralph Gross, and Jianbo Shi. Silhouette-based Human Identification from Body Shape and Gait. In *2002 Intl' Conference on Face and Gesture*, pages 351–356, October 2002.

[21] Ross Cutler and Larry Davis. View-based Detection and Analysis of Periodic Motion. In *International Conference on Patern Recognition*, page SA14, August 1998.

[22] Ross Cutler and Larry Davis. Robust periodic motion and motion symmetry detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2000.

[23] Ross Cutler and Larry Davis. Robust Real-Time Periodic Motion Detection, Analysis, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(2):129–155, 2000.

[24] James Davis, Aaron Bobick, and Whitman Richards. Categorical Representation and Recognition of Oscillatory Motion Patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635, June 2000.

[25] James W. Davis and Stephanie R. Taylor. Analysis and Recognition of Walking Movements. In *International Conference on Pattern Recognition*, pages 315–318, August 2002.

[26] Fernando De la Torre and Michael J. Black. Robust parameterized component analysis: theory and applications to 2D facial appearance models. *Computer Vision and Image Understanding*, 91(1/2), 2003.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[28] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. IEEE Int. Conference on Computer Vision and Pattern Recognition*, volume 2, pages 126–133, July 2000.

[29] J. Deutscher, B. North, B. Bascle, and Andrew Blake. Tracking through Singularities and Discontinuities by Random Sampling. In *Proc. IEEE International Conference on Computer Vision (2)*, pages 1144–1149, 1999.

[30] James Elding. Spatio-Temporal Similarity Measures. http://www.cs.ualberta.ca/ elding/. presentation.

[31] Helmy Eltoukhy and Khaled Salama. Multiple Camera Tracking. http://ise.stanford.edu/2002projects/ee392j/project.html, 2002. Digital Video Processing (EE392J) Student Final Project.

[32] Ronan Fablet and Michael J. Black. Automatic detection and tracking of human motion with a view-based representation. May 2002.

[33] Jeffrey P. Foster, Mark S. Nixon, and Adam Prugel-Bennett. New Area Based Metrics for Automatic Gait Recognition. In *Proceedings of British Machine Vision Conference*, pages 233–242, 2001.

[34] Scott Gaffney and Padhraic Smyth. Trajectory Clustering with Mixtures of Regression Models. In *Knowledge Discovery and Data Mining*, pages 63–72, 1999.

[35] Aphrodite Galata, N. Johnson, and D. Hogg. Learning Spatio-temporal Models of Human Behaviour. In *Proc. Spatial Temporal Modelling and its Applications (LASR)*. Leeds University Press, 1999.

[36] P. R. Giaccone and G. A. Jones. Spatio-Temporal Approaches to the Computation of Optical Flow. In *Proceedings of the British Machine Vision Conference*, pages 420–429, September 1997.

[37] M. A. Giese and T. Poggio. Synthesis and Recognition of Biological Motion Patterns Based on Linear Superposition of Prototypical Motion Sequences. In *Proceedings of the IEEE Workshop on Multi-View Modeling and Analysis of Visual Scene*, pages 73–80, June 1999.

[38] M. A. Giese and T. Poggio. Quantification and classification of locomotion patterns by spatio-temporal morphable models. In *Third IEEE Workshop on Visual Surveillance*, pages 27–36, July 2000.

[39] Kristen Grauman, Gregory Shakhnarovich, and Trevor Darell. A Bayesian Approach to Image-Based Visual Hull Reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR '03)*, volume I, pages 187–194, June 2003.

[40] Hayit Greenspan, Jacob Goldberger, and Arnaldo Mayer. Probabilistic Space-Time Video Modeling via Piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, March 2004.

[41] R. Gross and J. Shi. The CMU Motion of Body (MoBo) Database. Technical Report CMU-RI-TR-01-18, Carnegie Mellon University, June 2001.

[42] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[43] Bernd Heisele. Motion-based Object Detection and Tracking in Color Image Sequences. In *Fourth Asian Conference on Computer Vision*, pages 1028–1033, 2000.

[44] Alexander T. Ihler, Erik B. Sudderth, William T. Freeman, and Alan S. Willsky. Efficient multiscale sampling from products of Gaussian mixtures. In *Advances in Neural Information Processing Systems 16 (NIPS)*, 2003.

[45] Gareth James and Trevor Hastie. Functional Linear Discriminant Analysis for Irregularly Sampled Curves. *Journal of the Royal Statistical Society, Series B*, 63:533–550, 2000.

[46] Odest Chadwicke Jenkins and Maja J Matarić. Automated Modularization of Human Motion into Actions and Behaviors. Technical Report CRES-02-002, USC Center for Robotics and Embedded Systems, September 2002.

[47] G. Johansson. Visual motion perception. *Scientific American*, 232(6):75–80, 85–88, 1975.

[48] Todd R. Johnson, Hongbin Wang, Jiajie Zhang, and Yue Wang. A Model of Spatio-Temporal Coding of Memory for Multidimensional Stimuli. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*, pages 506–511. Lawrence Erlbaum Associates, 2002.

[49] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. In *2nd Int. Conf. on Automatic Face- and Gesture-Recognition*, pages 38–44, October 1996.

[50] Masakatsu Kourogi and Takeshi Kurata. Personal Positioning based on Walking Locomotion Analysis with Self-Contained Sensors and Wearable Camera. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 103–112, October 2003.

[51] N. Krahnstoever, M. Yeasin, and R. Sharma. Towards a Unified Framework for Tracking and Analysis of Human Motion. In *IEEE Workshop on Detection and Recognition of Events in Video (EVENT'01)*, pages 47–54, July 2001.

[52] Pierre Lane, K. Steven Knudsen, and Michael Cada. Moving Object Trajectory Estimation Using an Optical Fourier Processor. In *ICAPT98, SPIE Proceedings*, volume 3491, pages 939–942, July 1998.

[53] James J. Little and Jeffrey Boyd. Describing Motion For Recognition. In *SCV95*, page 5A Motion II, 1995.

[54] James J. Little and Jeffrey Boyd. Recognizing People by Their Gait: The Shape of Motion. *Videre*, 1(2):1–32, 1998.

[55] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.

[56] G.J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

[57] Anurag Mittal and Larry S. Davis. M2Tracker: A Multi-view Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo. In *ECCV(1)*, pages 18–36, 2002.

[58] Anurag Mittal, Liang Zhao, and Larry S. Davis. Human Body Pose Estimation Using Silhouette Shape Analysis. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)*, pages 263–270, 2003.

[59] Mirco Nanni. Distances for Spatio-temporal clustering (Extended Abstract). citeseer.nj.nec.com/498872.html.

[60] S. A. Niyogi and E. H. Adelson. Analyzing and Recognizing Walking Figures in XYT. In *IEEE Proc. Computer Vision and Pattern Recognition*, pages 469–474, June 1994.

[61] S. A. Niyogi and E. H. Adelson. Analyzing gait with spatiotemporal surfaces. In *IEEE Workshop on Nonrigid and Articulated Motion*, pages 64–69, 1994.

[62] John Oliensis. A Critique of Structure-from-Motion Algorithms. *Computer Vision and Image Understanding: CVIU*, 80(2):172–214, 2000.

[63] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie. Learning and tracking cyclic human motion. *Advances in Neural Information Processing Systems*, (13):894–900, 2001.

[64] Dirk Ormoneit, Hedvig Sidenbladh, Michael J. Black, Trevor Hastie, and David J. Fleet. Learning and Tracking Human Motion Using Functional Analysis. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000.

[65] Torsten Radtke and Volker Zerbe. Tracking of Dynamic Objects Based on Optical Flow. In *Proceedings of International Conference on Intelligent Multimedia and Distance Education*, pages 20–24, June 2001.

[66] Deva Ramanan and David A. Forsyth. Automatic Annotation of Everyday Movements. In *Proc. Of Neural Information Processing Systems (NIPS)*, 2003.

[67] Cen Rao, Mubarak Shah, and Tanveer Syeda-Mahmood. Action Rectionition based on View Invariant Spatio-temporal Analysis. In *ACM Multimedia 2003*, November 2003.

[68] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *Intl. J. of Computer Vision*, 50(2), 2003.

[69] Jens Rittscher and Andrew Blake. Classification of Human Body Motion. In *Proc. IEEE International Conference on Computer Vision (2)*, pages 634–639, 1999.

[70] John F. Roddick, K. Hornsby, and Myra Spiliopoulou. An Updated Bibliography of Temporal, Spatial and Spatio-Temporal Data Mining Research. In *Lecture Notes in Artificial Intelligence*, pages 147–16. 2001.

[71] Romer Rosales and Stan Sclaroff. Trajectory Guided Tracking and Recognition of Actions. Technical Report TR BU-CS-99-002, Boston U., September 1999.

[72] Payam Saisan and Alessandro Bissacco. Image-based modeling of human gaits with higher-order statistics. In *Proc. of the Intl. Workshop on Dynamic Scene Analysis*, June 2002.

[73] Steven M. Seitz and Charles R. Dyer. Affine Invariant Detection of Periodic Motion. In *ICVPR*, pages 970–975. IEEE, June 1994.

[74] Steven M. Seitz and Charles R. Dyer. Cyclic motion analysis using the period trace. In Mubarak Shah and Ramesh Jain, editors, *Motion-Based Recognition*, pages 61–85. Kluwer Academic Publishers, 1997.

[75] Mubarak Shah, Krishnan Rangarajan, and PingSing Tsai. Motion Trajectories. *IEEE Transactions on System Man and Cybernetics*, 23(4), July/August 1993.

[76] Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. Computing Gaussian Mixture Models with EM using Equivalence Constraints. In *Proc. Of Neural Information Processing Systems (NIPS)*, 2003.

[77] Cristian Sminchisescu. *Three-Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*. PhD thesis, GRAVIR - IMAG - INRIA, July 2002.

[78] Yang Song, Luis Goncalves, Enrico Di Bernardo, and Pietro Perona. Monocular perception of biological motion - detection and labeling. In *Proc. of 7th International Conferences on Computer Vision*, pages 805–812, September 1999.

[79] Yang Song, Luis Goncalves, Enrico Di Bernardo, and Pietro Perona. Monocular Perception of Biological Motion in Johansson Displays. *Computer Vision and Image Understanding*, 81(3):303–327, 2001.

[80] Yang Song, Luis Goncalves, and Pietro Perona. Unsupervised Learning of Human Motion Models. In *Advances in Neural Information Processing Systems 14*, December 2001.

[81] Yang Song, Luis Goncalves, and Pietro Perona. Unsupervised Learning of Human Motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(7):814–827, 2003.

[82] Robert Tibshirani. Principal Curves Revisited. *Statistics and Computing*, 2:183–190, 1992.

[83] Michael E. Tipping and Christopher M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

[84] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *Int. Journal of Computer Vision*, 9(2):137–154, 1992.

[85] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Learning Non-Rigid 3D Shape from Video. In *Proc. Of Neural Information Processing Systems (NIPS)*, 2003.

[86] D. Tran, M. N. Do, M. Wagner, and T. Van Le. A proposed decision rule for speaker identification based on a posteriori probability. In *Proc. of Speaker Recognition and Its Commercial and Forensic Applications*, pages 85–88, April 1998.

[87] Ping-Sing Tsai, Mubarak Shah, Katherine Keiter, and Takis Kasparis. Cyclic motion detection. Technical Report CS-TR-93-08, Computer Science Dept, University of Central Florida, 1993.

[88] PingSing Tsai, Mubarak Shah, Katharine Keiter, and Takis Kasparis. Cyclic Motion Detection for Motion Based Recognition. *Pattern Recognition*, 27(12), December 1994.

[89] C. J. Veenman, M. J. T. Reinders, and E. Backer. A Composite Model and Algorithm for Motion Correspondence. In *Proceedings of the 6th annual conference of the Advanced School for Computing and Imaging*, June 2000.

[90] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving Motion Correspondence for Densely Moving Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:54–72, January 2001.

[91] C. J. Veenman, M. J. T. Reinders, and E. Backer. Establishing Motion Correspondence using Extended Temporal Scope. *Artificial Intelligence*, 145:227–243, April 2003.

[92] C.J. Veenman, E.H. Hendriks, and M.J.T. Reinders. A Fast and Robust Point Tracking Algorithm. In *Proc. of the Fifth IEEE International Conference on Image Processing*, pages 653–657, October 1998.

[93] Michail Vlachos, Dimitrios Gunopulos, and George Kollios. Robust Similarity Measures for Mobile Object Trajectories. In *DEXA 2002, 5th International Workshop Mobility in Databases and Distributed Systems*, September 2002.

[94] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories, 2002.

[95] Liang Wang, Weiming Hu, and Tieniu Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, March 2003.

[96] Lior Wolf and Amnon Shashua. Learning over Sets using Kernel Principal Angles. *JMLR*, 4:913–931, October 2003.

[97] Lei Xu and Michael I. Jordan. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8(1):129–151, 1996.

[98] Yaser Yacoob and Michael J. Black. Parameterized Modeling and Recognition of Activities. *Computer Vision and Image Understanding: CVIU*, 73(2):232–247, 1999.

[99] Yaser Yacoob and Larry S. Davis. Learned Temporal Models of Image Motion. In *ICCV-98*, pages 446–453, 1998.

[100] Yaser Yacoob and Larry S. Davis. Learned Models for Estimation of Rigid and Articulated Human Motion from Stationary or Moving Camera. *Int. Journal of Computer Vision*, 36(1):5–30, 2000.

[101] ChewYean Yam, Mark S. Nixon, and John N. Carter. Automated person recognition by walking and running via model-based approaches. *Pattern Recognition*, 37(5):1057–1072, May 2004.

[102] Jang-Hee Yoo and Mark S. Nixon. Markerless Human Gait Analysis via Image Sequences. In *Proceedings of International Society of Biomechanics XIXth Congress*, 2003.

[103] Tao Zhao and Ram Nevatia. 3D Tracking of Human Locomotion: a Tracking as Recognition Approach. In *International Conference on Pattern Recognition*, August 2002.

Izjava

Izjavljam, da sem magistrsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Aleša Leonardisa. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.